KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Active Sensing for Touch-based Object Localisation

Niccoló Tosi

Dissertation presented in partial
fulfillment of the requirements
for the degree of Doctor in
Engineering Science

March 2015

Active Sensing for Touch-based Object Localisation

Niccoló TOSI

Examination committee:
Prof. dr. ir. Jean Berlamont,
preliminary defence chairman
Prof. dr. ir. Willy Sansen,
public defence chairman
Prof. dr. ir. Herman Bruyninckx,
supervisor
Dr. ir. Erwin Aertbeliën
Prof. dr. ir. Joris De Schutter
Prof. dr. ir. Luc De Raedt
Dr. Alain Micaelli (CEA LIST)
Prof. Vincent Padois (UPMC)
Prof. Anders Robertsson (Lund University)

Dissertation presented in partial
fulfillment of the requirements
for the degree of Doctor
in Engineering Science

March 2015

Acknowledgments

This dissertation is the result of four years of hard work carried out with the support of many people to whom I will always be grateful. Here, I wish to express all my gratitude for their invaluable contribution.

First of all, I wish to thank my supervisors, Olivier David and Herman Bruyninckx, for their support, enthusiasm, and constructive criticism. Together, we have been a great team. Their mentorship has been fundamental from both a professional as well as a personal point of view.

I wish to thank Yann Perrot, who believed in this PhD project from the moment Olivier and I walked into his office with our doctoral proposal. Thanks to all my colleagues from CEA and KU Leuven for their support and the many scientific discussions, especially those in front of the coffee machine. Many thanks to my fellow PhD students who read my articles, as well as this thesis, for helping me improve both content and form: Julie Dumora and Davide Nitti. Thanks to Enea Scioni, for sharing his software know-how with admirable patience. Thanks to my official Dutch translators: Wilm Decré, Maxim Vochten and Kevin Tanghe.

Thanks to Franck Geffard for supporting my PhD application and for his valued help with TAO. Thanks to Xavier Lamy for all the time spent in the lab, shoulder-to-shoulder, always with a smile. Thanks to Marc Itchat and Benoit, for their help in setting up the robot experiment. Thanks to Jérémie Le Garrec for driving me through XDE. Thanks to Fabien Dionnet, for sharing long coding hours and not making me feel lonely as a Linux user. Thanks to Brian Garcia for thoroughly proofreading my articles and hosting me in his apartment.

A special thanks to my mother and father, Maria and Stefano, and to my grandmothers, Giuliana and Lore, for their support throughout my studies and research activity. Lastly, but certainly not least, thanks to Giorgia, my fiancée, who believed in me, encouraged me, and always stood by my side through the good and bad moments. Without her sacrifice and support, I could not have

completed this journey.

Abstract

This doctoral thesis improves the way robots exploit their force and tactile sensing capabilities in order to **localise** objects in their surrounding, with specific focus on industrial relevance, that is **minimising execution time** and working with items of **real-world geometric complexity**.

Driven by the need to analyse the decision-making process occurring during localisation operations, we present a **test carried out to observe human approach to touch-based localisation tasks**. Common behaviour patterns among the subjects were observed, such as following the same sequence of actions and being attracted by specific features of the object.

Generalising from relevant literature works, an **Active Sensing Task model** is elaborated based on five modelling concepts: configuration space, information space, action space, inference scheme and action-selection scheme. Concrete examples from the literature conforming to the model are presented. In this modelling context, an action-selection **algorithm** named *act-reason* is developed providing a configurable trade-off between information gain, cost of motion-execution and computation.

Inspired by the results of the human experiment, **DOF Decoupling** is introduced as a **methodology** to describe localisation applications as coordination of active-sensing tasks with different configuration spaces. This allows the robot to reduce the complexity of the problem, thus improving the online-feasibility of the localisation application. Previous touch-based object-localisation works conform to this methodology, and concrete examples from the literature are provided.

A 3DOF rectangle localisation on top of a table is carried out to validate the effectiveness of DOF Decoupling in **reducing the complexity of the application**. Specifically, the scheme observed in the human experiment is applied to decouple the whole application into an initial 2DOF table exploration task and a 3DOF localisation task after the first contact. Since the complexity

of the configuration scales exponentially in the number of degrees of freedom and affects that of the inference scheme, pose estimation in 2DOFs is significantly less computationally expensive than in 3DOFs. Intuitively, to speed up the computation, DOF Decoupling allows the robot to set the pose estimator at the most appropriate level of resolution, *i.e.* coarse with high uncertainty and fine with low uncertainty. Examples with initial uncertainty up to [0.8m,0.8m,180deg] are presented. The time required to accomplish the localisation with and without decoupling is compared showing a significant improvement in the former case.

To assess the robustness of our measurement model and inference scheme in an industrial context, we localise **objects up to industrial-relevance complexity** using a Staubli RX90 robot equipped with a spherical end effector coupled with a force-torque sensor.

Case-study implementations are presented to show how the *act-reason* algorithm can be effectively adopted to reduce the time to localise an object. In particular, in the proposed examples, the allocated time to reason and act is a function of the current uncertainty. Improvements upon naive strategies are reported in terms of the total time required by the application.

Beknopte Samenvatting

Dit proefschrift verbetert de manier waarop robots hun kracht- en tactiele sensoren kunnen gebruiken om objecten in hun omgeving te lokaliseren. Specifieke aandacht gaat hierbij naar de industriële relevantie: het minimaliseren van de takttijden en het werken met objecten met een industrieel relevante geometrische complexiteit.

Om het besluitvormingsproces, dat optreedt tijdens lokalisatietaken, te analyseren, stellen we een test voor om de menselijke aanpak voor deze taken te observeren. Er werden hierbij gemeenschappelijke onder de proefpersonen waargenomen, zoals het volgens dezelfde sequentie uitvoeren van handelingen of het worden aangetrokken door specifieke kenmerken van het object.

Startend vanuit een veralgemening van relevante literatuur, is een active sensing taakmodel uitgewerkt op basis van vijf modeleerconcepten: de configuratieruimte, de informatieruimte, de actieruimte,het inferentieschema en het actie-selectieschema. Voor deze modellen geven we concrete voorbeelden vanuit de literatuur. In deze modelleercontext, wordt een actie-selectie-algoritme met de naam handel-redeneer ontwikkeld dat een configueerbare afweging toelaat tussen informatiewinst, kost van bewegingsuitvoering en rekenkracht.

Geïnspireerd door de resultaten van de experimenten uitgevoerd door proefpersonen wordt vrijheidsgraadontkoppeling geïntroduceerd als een methode voor het beschrijven van lokalisatietoepassingen als coördinatie van active sensing-taken met verschillende configuratieruimtes. Hierdoor kan de robot de complexiteit van het probleem verminderen en daardoor de haalbaarheid van de lokalisatietoepassing voor online gebruik verbeteren. Eerder werk rond tastgebaseerde objectlocalisatie zijn in overeenstemming met dit model en het proefschrift geeft concrete voorbeelden uit de literatuur.

Een drie-vrijheidsgraden rechthoeklokalisatie op een tafel is uitgevoerd om de doeltreffendheid van vrijheidsgraadontkoppeling aan te tonen voor het verminderen van de complexiteit. Concreet wordt het waargenomen schema uit

de proefpersoonexperimenten gebruikt om de hele toepassing te ontkoppelen in een eerste twee-vrijheidsgraden tafelverkenningtaak en een tweede drie-vrijheidsgraden lokalisatietaak na het eerste contact. Aangezien de complexiteit van de configuratie exponentieel schaalt met het aantal vrijheidsgraden en tevens de complexiteit van het inferentieschema beïnvloedt, is de poseschatting in twee vrijheidsgraden significant minder rekenintensief dan in drie vrijheidsgraden. Intuïtief, om de berekening te versnellen, laat vrijheidsgraadontkoppeling toe dat de robot de poseschatter instelt op de meest geschikte resolutie., dit wil zeggen grof bij een hoge mate van onzekerheid en fijn bij een lage onzekerheid. Voorbeelden met aanvankelijke onzekerheid tot [0,8m, 0,8m, 180 graden] worden gegeven. De nodige tijd om de lokalisatie uit te voeren met en zonder ontkoppeling wordt vergeleken en is significant lager bij ontkoppeling.

Om de robuustheid van het meetmodel en inferentieschema in een industriële context te beoordelen, lokaliseren we objecten met een industrieel-relevante complexiteit met een Staubli RX90 robot voorzien van een bolvormig eindeffector gekoppeld aan een kracht-momentsensor.

Gevallenstudie-implementaties laten zien hoe het handel-redeneer-algoritme de tijd om een object te lokaliseren kan reduceren. In het bijzonder, voor de gegeven voorbeelden, is de toegewezen tijd om te redeneren en handelen functie van de huidige onzekerheid. Verbeteringen ten opzichte van naïeve strategieën worden aangegeven als de totale tijden nodig voor de toepassing.

Contents

List of Figures

List of Tables

Nomenclature

Symbols

$\{...\}$	:	set
$\|.\|$	:	matrix determinant
$\|\|.\|\|$	:	norm operator
α	:	contact-detection false negative rate
β	:	contact-detection false positive rate
δ	:	transition (process) noise
ϵ	:	measurement noise
γ_P	:	protection orientation (human experiment)
γ_T	:	orientation of the table w.r.t. the robot z axis (human experiment)
γ_{VB}	:	orientation of the v-block w.r.t. the robot z axis (human experiment)
Γ	:	maximum range for distance calculation
η	:	normalisation factor
κ	:	sweep thickness
$\lambda(.)$	:	matrix eigenvalues
$\xi(.)$	:	boolean intersection function
Π	:	localisation progress metric
ρ	:	radius of the bounding circle
Σ	:	covariance matrix
σ_P	:	contact-point measurement standard deviation error
σ_n	:	normal vector measurement standard deviation error
τ	:	trajectory
τ_c	:	trajectory up to contact point
ϕ	:	robot pose (frame) at the end of trajectory
ϕ_c	:	robot pose (frame) when in contact
ϕ_{end}	:	robot pose (frame) at the end of the trajectory

ψ	:	swept volume
χ	:	particles' set
$\bar{bel}$	:	conditional probability (belief) after transition in Bayes Filter
bel	:	conditional probability after measurement update in Bayes Filter
a	:	action
C	:	contact measurement
d	:	penetration depth
dim	:	number of dimensions of the solution space
$dist(.)$	:	Euclidean distance
$E[.]$	:	expectation
f_i	:	i-th face of the polygonal-mesh
f^*	:	polygonal face selected to represent the object when calculating the likelihood function
Fx	:	force signal along the x axis
Fy	:	force signal along the y axis
Fz	:	force signal along the z axis
$g(.)$	:	transition model
h	:	decision-making time horizon
H	:	probability (Shannon) entropy
IG	:	information gain
$k(.)$	:	Parzen window (kernel)
$m(.)$	:	measurement model
n_i	:	normal vector of the i-th face of the polygonal mesh
n_{int}	:	number of intersected rectangles
n_{tot}	:	number of total rectangles
n_a	:	number of actions
NC	:	no-contact measurement
obj_{ET}	:	ergonomy-test object
obj_{SR}	:	solid rectangle object
obj_{VB}	:	v-block object
$P(.)$	:	probability
$p1$	:	protection 1 (human experiment)
$p2$	:	protection 2 (human experiment)
r	:	reward
$S1$	:	1st subtask
$S2$	:	2nd subtask
$S3$	:	3rd subtask
Sol	:	size of the solution space
t	:	time

t_{alloc}	:	allocated time to evaluate and execute the next sensing action
t_c	:	computational time
$\bar{t}_{2nd}$	:	average time to second contact
$\bar{t}_{ent}$	:	average execution time with entropy-minimisation action selection
$\bar{t}_{rand}$	:	average execution time with random action selection
$\bar{t}_{tot}$	:	average time to execute the localisation task
res	:	resolution
$tr(.)$	:	matrix trace
x	:	random variable
$\mathbf{x}$	:	object pose
$x^{[i]}$	:	i-th particle
$\hat{\mathbf{x}}$	:	best-estimate pose
$w^{[i]}$	:	weight of the i-th particle
W	:	weights of the weighted covariance matrix
z	:	measurement
z_{CD}	:	contact-detection measurement
z_n	:	normal-vector measurement
z_P	:	contact-point measurement
z_{touch}	:	contact measurement (contact point and normal vector)

General Abbreviations

CEA	: Commissariat à l'Énergie Atomique et aux Énergies Alternatives
CF	: Contact Formation
DOF	: Degree of Freedom
EKF	: Extended Kalman Filter
FT	: Force-torque
KF	: Kalman Filter
MDP	: Markov Decision Process
POMDP	: Partially Observable Markov Decision Process
SLAM	: Simultaneous Localisation and Mapping
UKF	: Unscented Kalman Filter
XDE	: eXtended Dynamic Engine

Chapter 1

Introduction

This doctoral thesis improves the way robots exploit their force and tactile sensing capabilities in order to **localise** objects in their surrounding, with specific focus on industrial relevance, that is **minimising execution time** and working with items of **real-world geometric complexity**.

In this context, this thesis tackles three major research challenges, in increasing order of concreteness:

- **Methodological**: to find a **conceptual** approach to discuss the planning, sensing and control aspects at **the most effective level of abstraction** during all phases of a sensor-based robot task in uncertain environment.
- **Modelling**: to find **formal representations** of the aforementioned concepts allowing to develop planning, sensing and actions algorithms for touch-based localisation.
- **Algorithmic**: the development of a concrete action-selection **algorithm** providing a configurable trade-off between information gain, cost of motion execution and computation.

1.1 Motivation and Use Cases

Typical robotics applications, *e.g.* bin picking or object manipulation, require the robot to be able to place its end-effector in a given location within a specified tolerance. A reliable 3D model of the environment becomes a fundamental

building block for such operations. Vital to enhance the reliability of the 3D model, scene calibration is the act of estimating the pose (*i.e.* the position and orientation) of the objects present in the environment.

Built upon a CAD representation of the scene objects, the virtual model may be exploited, for instance, to assist the human operator while performing tele-manipulation tasks or to plan actions in cluttered environments avoiding undesired contacts. In order to perform calibration, different forms of sensing can be adopted including vision, laser, ultrasound and touch. However, such operation may become non-trivial and potentially time-consuming, depending on the harshness of the scene and the initial uncertainty.

Concretely, this doctoral project has been inspired by a problem occurring in tele-robotics operations for tunnel-boring applications in the framework of the Telemach project [13]. While excavating the tunnel, the cutter head was filled with bentonite, a fluid clay mixture used as lubricant and containment barrier. When the boring was stopped for maintenance, the bentonite was partially flushed out and replaced by air. This made the excavating chamber suitable for human intervention, even though, from a safety point of view, replacing bentonite with air is not a good solution due to the compressibility of the gas. The Telemach project focused on performing such maintenance operations (*e.g.* substituting a cutting head) using remotely-controlled robots with the aid of a 3D model of the scene to provide assistance to the human operator, *e.g.* with virtual guidance and multiple virtual-camera view points. However, this virtual model had to be calibrated using cameras embedded on the robot to align it reliably with respect to the actual scene. In some cases, due to the heavy presence of uniformly-coloured mud and rocks, the robot was not capable of identifying useful visual features and thus calibration became impossible. For the sake of clarity, Figure 1.1 illustrates a lab mock-up for the cutting-head insertion, together with the actual on-site scenario covered in soil. Further to the experienced calibration difficulties, there was the need of exploiting physical interactions with the environment in order to calibrate the model even in case of camera-unfriendly conditions.

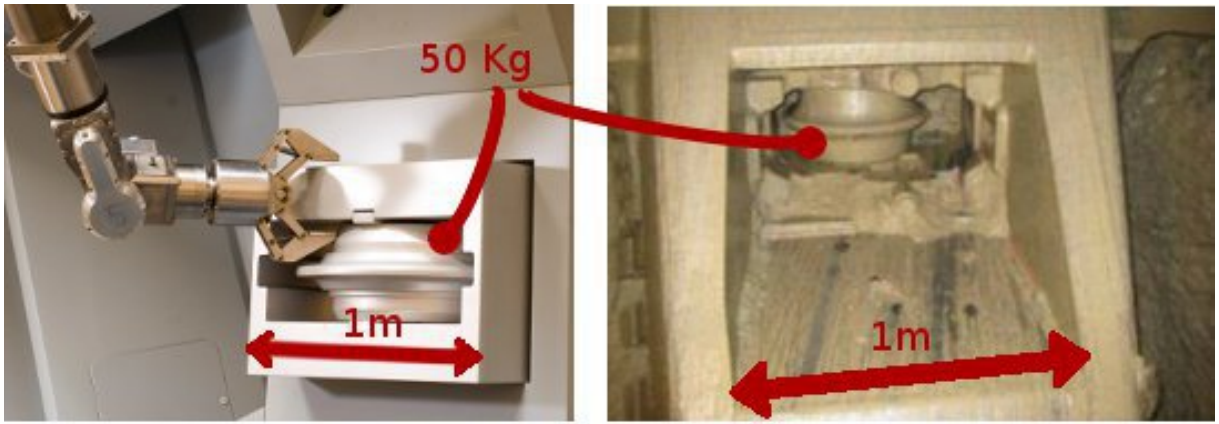

Figure 1.1: Example of tele-operation application for cutting-head substitution in tunnel boring operations. The image shows significant difference between lab and intervention scenario conditions. Due to the heavy presence of mud and water, camera-based model calibration was difficult.

In a more general sense, allowing the robot to explore and calibrate its workspace model through physical interaction is relevant for a number of applications where cameras operate in non-optimal conditions. For instance, during deep-sea operations where cameras struggle against bad lighting, shadows, bad contrast and water turbidity, or in industrial scenarios where the scene has a uniform stainless steel color and backscattered light becomes an issue.

1.2 Definition of Key Concepts

Inspired by the Telemach industrial scenario, this thesis focuses on force sensing as a reliable mean to explore the robot environment. In this context, a sensing primitive is comprised of a robot motion and the force-torque signals measured by a sensor coupled with the robot end effector. Generalising from this concrete case leads to the following definitions at **conceptual level**; the rest of the thesis text will go into more details about **models** and **algorithms**.

Def. 1.1. Task: *any controlled motion of the robot interpreting information collected through its sensory system in order to realise a planned change in the world.*
More specifically, control is responsible for realising a desired change in the world. Sensing collects information about the state of the world. Planning selects sensing and controls which are expected to realise desired changes in the world. In general, a task is composite, *i.e.* it may be composed by other tasks of lower level of abstraction. At conceptual level, we deliberately avoid the definition of concepts like "action" or "primitive task" since they only become relevant as soon as one introduces concrete formal representations (in other words "models") of tasks at a defined **level of abstraction**. If used without specific context, the term "application" is considered a synonym of task in this thesis.

Def. 1.2. Active Sensing Task: *task whose desired outcome is to gain information about the pose of the objects in the world.*
In the context of object localisation, one has to make concrete choices of models to represent the world, *i.e.* to desribe where the objects are in the environment. The choice of world representation influences the choice of the models to represent the controller, the sensing and the planning. In Chapter 3, a model-level definition of active-sensing tasks in the context of touch-based localisation will be presented.

Def. 1.3. Configuration space of a system: *set of the possible states of that system.*
In general, this definition applies to all the systems related to a task, *e.g.* world,

control, sensing and planning. Since this thesis focuses on object localisation, when used without explicit context, the term "configuration space" refers to the set of possible poses that the object can have in the world.

Def. 1.4. Degrees of Freedom (DOFs) of a system: *number of independent parameters to represent the system's configuration space.*
Throughout this thesis, the configuration spaces of the pose of the objects present 2, 3, and 6 DOFs, or combinations of these DOFs.

Def. 1.5. Complexity: *amount of resources required to describe a configuration space.*
Since planning, sensing and control are all working with the representation of the world, the complexity of the configuration space of the pose of the objects has a direct influence on the complexity of the representations and algorithms for planning, sensing and control. In general, complexity metrics are task dependant, since different systems and processes exploit different forms of resources. For instance, let us represent a configuration space discretising it with a uniform grid. The number of necessary nodes N can be used as complexity metric for the configuration space. Specifically, N depends on the DOFs (n_{DOFs}) composing the configuration space, on the grid resolution res (*i.e.* the distance between two adjacent nodes) and on the ranges (*i.e.* the difference between the maximum and the minimum value for each DOF). Hence, the complexity of the configuration space is that expressed in Eq. 1.1.

$$N = \prod_{i=1}^{n_{DOFs}} \frac{range_i}{res_i}. \tag{1.1}$$

Concretely, throughout this thesis, the world will be represented by a **geometric model** composed of polygonal objects. Their configuration space, *i.e.* their pose in the world, will be expressed by DOFs defined with respect to the robot frame of reference. In a touch-based active-sensing context, the **model-level** definition of an action is here provided.

Def. 1.6. Action: *a controlled robot trajectory which provides observations of the pose of the objects in the world.*
The observations collected during an action depend on the control and the sensing apparatus.

1.3 Research Objectives

Focusing on the industrial requirement of performing **fast and reliable** scene calibration, this doctoral project aims at coping with the **curse of**

dimensionality related to action selection and pose estimation in case of high initial uncertainty.

Among different touch-based sensors, it is chosen to work with **force sensors**. They represent the most suitable solution to work in the harsh environments identified as potential use cases.

With respect to the prior art, the problem of action selection is studied aiming at finding solutions which **minimise the total time** required to perform the calibration in a task-oriented fashion explicitly trading off information, execution time and computation cost.

Finally, this project aims at implementing real-life examples of object localisation with a robot manipulator interacting with objects up to **industrial complexity**.

1.4 State of the Art

This Section is dedicated to position this doctoral research with respect to the state of the art which collects findings from different domains, including mechatronics, computer science and probabilistic inference.

Subsection 1.4.1 presents an overview of the available **sensors** which may be adopted to collect contact measurements. Subsection 1.4.2 summarises the prior art on how the collected contact data can be used to **infer** object pose. Subsection 1.4.3 presents a literature review on active sensing.

1.4.1 Touch Sensing Devices

The class of touch-based sensors is comprised of a large set of transducers which sense information by physical contact with the surrounding environment. This subsection briefly presents the state of the art on this type of devices, focusing on their application for solid-object localisation.

- **Tactile sensors:** typically based on strain-gauge [12], piezo-electric [39] or capacitive [68] technology, they are used to measure shape, texture, softness, temperature, vibration or shear and normal forces. In the framework of object localisation, they provide an "image" of the entities in contact with robot structure, covered with a "skin" of tactile arrays. In literature, tactile sensors have been used for object localisation mainly for grasping applications [28, 30, 52]. Their maximum measurable pressure is

in the order of 200 KPa [45]. In case of limited application surfaces (*e.g.* $10cm^2$), the range of measurable forces is significantly low if compared to the type of interactions experienced in the Telemach project [13].

- **Force-torque (FT) sensors:** these probes generate an electrical signal proportional to the forces and torques applied. Historically based on strain-gauge technology, a FT load cell is usually comprised of four strain gauges mounted in a Wheatstone bridge configuration for each axis. More recently, a piezoelectric-based resonant probe has been proposed [10] as alternative device which relies on the frequency shift that changes according to the applied load. Its estimated production cost is interestingly low, but its application in robotics is still under development. Usually, in the context of object localisation, FT sensors are coupled with a spherical end effector [21, 61, 57, 65] which allows the robot to estimate contact point and normal vector through the pseudo-contact technique [38]. Alternatively, the geometric properties of the touched surface can be estimated through continuous compliant motion [42]. Measurable forces range up to 10^5N with different resolutions depending on the quality (and price) of the sensor.

- **Whiskers:** Inspired by animals' ability to use hairs for sensing [3], these are binary probes used to detect contact along a rigid or semi-rigid structure attached to the robot [37]. Examples in robotics literature present mobile [34] and object-exploration [58] applications. In recent years, they have been used to perform grid-based SLAM in small environments [19]. The principal advantage of whiskers is their cheap design, however an effective sensor system may require the integration of several probes.

In summary, several touch-based probes are commercially available. While tactile sensors provide fine local measurements, their limited force range makes them not suitable for the harsh environments identified as use cases for the current project. Compared to whiskers, force sensing allows the robot to estimate the normal vector of the touched surface with a single contact point and are therefore adopted in this work.

1.4.2 Touch-based Localisation

In the early 80s, 3DOF and 6DOF polyhedral-object localisation was performed using interpretation trees to find the hypothesis on the object pose that best fitted the available tactile data [24, 27]. This approach relies on the assumption that the tactile data fully constrain the position and orientation of the object.

The proposed algorithm considers contact position and surface normals, pruning those hypotheses which are not consistent with the collected measurements. With such formulation, the computational complexity is proportional to the number of configuration hypotheses to test. However, the interpretation-tree algorithm does not take into account uncertainty and assumes perfect measurement and geometric models.

Probabilistic schemes were then developed to perform object localisation through Bayesian inference [14, 21, 57]. The pose of the object is modelled as an unobserved Markov process and is estimated by building a posterior probability distribution over the possible states, conditioned on the robot control actions and the collected measurements [28, 30, 52]. In literature, such posterior is also referred to as belief [63]. Bayesian filtering allows to infer the pose of the object processing control and measurement data in a recursive fashion over time. In this framework, robot motions are observable but are not deterministic processes, *i.e.* their outcomes is not known *a priori*, and measurements are affected by noise. They are both represented as stochastic processes with associated probability distributions, usually named transition model and measurement model, respectively. Essentially, a Bayes filter is composed of two basic steps: prediction and update. In the prediction step, a probability distribution over the state is built based on the prior belief and the transition model that accounts for the control performed. In the update step, after a measurement is collected, the predicted distribution is multiplied by the measurement probability. For a more in-depth description of the Bayes filter and its implementations we refer to [63].

In general, Bayesian techniques represent a powerful mean for state estimation incorporating model-based information and accounting for system noise in an explicit way. However, in practical cases, they may require significant computational resources and memory space at run time. For this reason, many of the available Bayesian filtering models make simplifying assumptions on the probability functions used to describe the transition process and the measurement model. Specifically, the Kalman Filter [35] (KF) assumes linear motion and measurement models biased by white noise. This widely-adopted parametric estimator is optimal in the sense that it guarantees the solution to minimise the mean square error of the estimated parameters. The relaxation of the assumption of linear motion and measurement models led to the definition of the Extended Kalman Filter [25] (EKF) and the Unscented Kalman Filter [67] (UKF) coping with a wider set of real-life systems but paying the price of not guaranteeing optimality. Other non-parametric inference models, such as grid-based methods, Histogram Filter and Particle Filter [17], rely on discretisation to approximate the posterior distribution, making no assumption about its nature. However, they may require a significant computational effort since

the probability on the state given the measurements has to be calculated N times, where N is the number of samples that discretise the configuration space. Nevertheless, thanks to their flexibility and general applicability, these recursive state-estimation models are adopted to solve a vast number of problems, including applications for mobile robots [64], autonomous underwater vehicles [46], unmanned air vehicles [59] and touch-based localisation [57, 30, 28]. This thesis makes no difference in this sense. For the sake of clarity, more details about Bayesian Filtering are provided in the Appendix 7.3.

As for Bayesian inference applied to object localisation with local information, in 1996, De Geeter *et al.* [15] studied the problem of localising solid objects using ultrasonic and infra-red sensors. Objects were represented as a set of constrained analytical primitives (*e.g.* planes, lines and cylinders). The location of the object is represented by a mean value of the state vector and a covariance matrix, thus modelling the uncertainty on the pose as a multivariate Gaussian distribution in 3DOFs. In the proposed methodology, estimation was achieved using a the Smooth Constrained Kalman Filter (SCKF), which consists of an Extended Kalman Filter that enforces the object geometrical constraints in a relaxed fashion in order to avoid linearisation errors which prevent the algorithm from converging to the true value. The model was validated with experiments on a rectangular plate. A pre-defined feature point on the plate surface was localised with a standard deviation error of 3mm starting from an uncertainty modelled as normally distributed with diagonalised covariance matrix with $diag(\Sigma) = [0.042m^2, 0.041m^2]$.

In 2001, Gadeyne *et al.* [21] performed force-based cube localisation in 3DOFs. A grid-base method was chosen to represent the posterior distribution by uniformly sampling the configuration space. To estimate the pose of the fixed object, the robot processed measurements collected through a force-torque sensor coupled with a spherical end effector. In the presented experimental implementation, the initial uncertainty was [0.3m,0.3m,90deg], with a grid resolution of 0.002m and 2deg. In 2005 [22], they adopted sequential Monte Carlo sampling to simultaneously estimate contact-state formations and geometrical parameters during cube-in-corner operations. This work builds upon the results presented by Lefevbre *et al.* [42], where the same cube-in-corner operation was decomposed into a sequence of compliant contact formations (CFs), each of them described as a number of elementary formations, *e.g.* vertex-plane and edge-plane. This allowed the automatic generation of measurement equation for each CF.

In 2006, Petrovskaya *et al.* [57] performed 6DOF force-based localisation with uncertainty range of 400mm in position and 360 deg in orientation. A spherical end effector coupled with a force-torque sensor allowed to estimate the contact point and the normal vector of the touched surface. Both measurements were supposed to be affected by Gaussian noise. For the inference, they used an

enhanced version of the Particle Filter named Scaling Series. In this thesis, we adopt the same measurement model. The authors present results obtained in a box-localisation case application. In a later paper [56], the model was applied to localise a set of more complex objects modelled as a polygonal mesh with up to 100 faces.

In recent years, Hsiao [31] performed object localisation and grasping using a three finger hand equipped with a 6-axis force-torque sensor on each finger and binary contact sensors on the tips and the palm. A similar set-up was adopted by Hebert [28]. Both works focused on decision making and will be analysed in more detail in Subsection 1.4.3.

In summary, Bayesian methods are well established techniques for state estimation in the context of touch-based localisation. Prior works have defined measurement models to encode the information collected during free-space motions (no-contact sensing) and through physical interaction with tactile sensors, force sensors and boolean contact probes. Though powerful, Bayesian estimation tends to become computationally cumbersome when the assumptions of linear models and Gaussian noise are not acceptable, and when the complexity of the configuration space increases. To keep the problem tractable, an upper bound on such complexity was established *de facto*, thus allowing the robot to select actions online. In this context, we refer to online action-selection when the decisional process lasts no longer than the time required to execute the chosen action.

1.4.3 Active Sensing

Previous related works analysed the problem of actively sensing the environment for 3DOF self-localisation tasks. In [20], Fox *et al.* applied Markov localisation to mobile robotics in order to infer position inside a structured office environment using laser and ultra-sound sensors. A cost-vs-utility function was introduced to select the best movements and sensing actions to perform. Costs were represented by time and energy spent, and utility was represented by the expected decrease of uncertainty evaluated through the Kullback Leibler Divergence (D_{KL}) [40], *i.e.* the non-symmetric measure of the difference between the current posterior probability distribution and the expected distribution after the sensing action. Further to that work and focusing on reducing the sensor resources required, the same task was performed using angular and linear odometers, a compass and a contact sensor, an angular odometer and a contact sensor [54]. More recently, the same operation was accomplished only using a clock and a contact sensor [18].

The problem of choosing *where-to-sense-next* with proximity probes to localise solid objects was studied by De Geeter *et al.* in [14]. Under Gaussian assumptions for the joint distribution over the pose of the object, a comparison between covariance-based methods was presented. The authors opted for the so-called tolerance-weighted L-optimal design, *i.e.* a decision-making scheme where actions are selected as to minimise the weighted trace of the covariance matrix. This specifies a sequence of actions that depends on the information needed for task, encoded in the weighting matrix.

Active force sensing to perform cube-in-corner tasks with autonomous motions was studied in [43] by Lefebvre *et al.*. The operation was decoupled into a compliant sequence of contact formations each composed of an elementary contact state, *e.g.* corner-to-face, edge-to-face and face-to-face, as depicted in Figure 1.2. A topological representation of the possible sequences of CFs and compliant transitions was stored in the so-called CF Graph. For each CF, the set of observable parameters was defined by linearizing the measurement equation and by calculating the Fisher Information Matrix I_f. The eigenvalue decomposition of $I_f = U\Theta U^T$ indicated the expected information gain on the diagonal of Θ for the directions collected in the columns of U^T. Then, the decision making was formulated as an optimization process decoupled into a requirement for the CF sequence and a requirement for the active sensing motions in each CF. Results from this work proved that planning the whole sequence of actions for the cube-in-corner application is computationally expensive and may not be performed online.

In 2009, Hsiao [32] performed object localisation and grasping in 3DOFs using tactile sensors installed on a three-finger hand. The authors focused on the decision-making side of the problem adopting the Partially Observable Markov Decision Process [9] (POMDP) scheme to maximize some gain metric over a finite-horizon sequence of motions. This framework is a generalisation of the Markov Decision Process (MDP) [5], where decision making is formalised taking into account actions with stochastic outcomes. POMDP builds upon MDP relaxing the assumption of full-observability of the state, thus maintaining a probability distribution function over the set of possible states. Further information on MDP and POMDP is provided in the Appendix. At each time step, the robot had to select an action composed of a three-finger trajectory defined off-line relatively to the object geometry, detecting contact and measuring the interaction forces with finger-tip tactile sensors. The presented experimental results were carried out with the time horizon set to 2, that is deciding what to do now taking into account the expected gain over the next two actions. The configuration space representing the possible poses of the object was discretised as a grid with 2.4 x 10^4 nodes. Then, Bayesian inference was adopted to update the posterior distribution conditioned on the measurement collected during

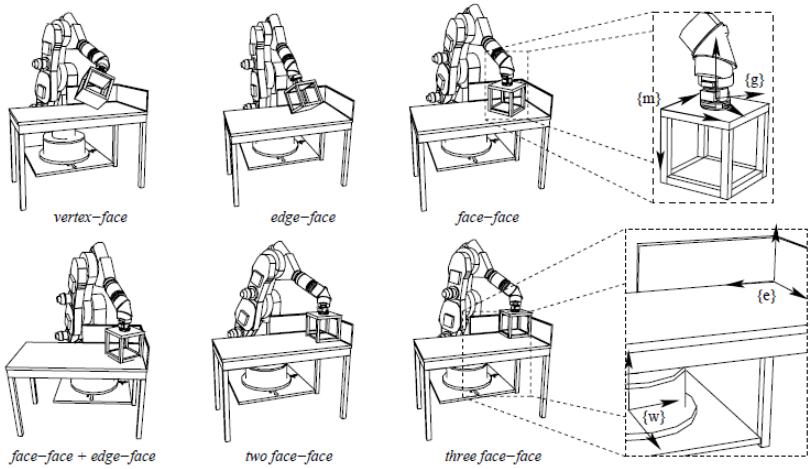

Figure 1.2: Active-sensing for a cube-in-corner operations. The decision-making was comprised of a requirement for the discrete sequence of contact formations and a requirement for the continuous active sensing motions in each of them.

both free-space motions and contact situations. In the former case, the volume swept by the robot was used to assign low probability to those configurations which would have caused a contact along the trajectory. In the latter case, information is processed using a similar model as in [57]. In [29], instead of an information metric, the probability of grasping success as a function of the pose uncertainty was used as reward function for making decisions. Even in case of low initial uncertainty [5cm,5cm,0.5rad], the time required for the decision making was significant, approximately an order of magnitude greater than the time to execute the chosen action.

Taguchi *et al.* [62] studied the problem of selecting the next best action for touch-based registration measuring information gain with probability entropy, which is a metric to compute the dispersion of a probability distribution even in case of non-Gaussian conditions derived from information theory [60] (further details are provided in the appendix). The configuration space was discretised and a Particle Filter was used for the inference. They compared different methods to compute the entropy function approximating the posterior distribution by a single gaussian, a kernel distribution and the weights of a running Particle Filter.

In 2012, Hebert *et al.* [28] performed object localisation using a hand equipped with tactile sensors that allowed the robot to estimate the contact point on each

finger and the normal vector of the touched face. Actions were selected as to maximise the immediate information gain reward, measured through the D_{KL}. Inference was performed by means of a Histogram Filter and the measurement models were equivalent to those adopted in [31]. As in previous examples, actions were defined off-line by the task programmer relatively to the geometry of the object with the hand equipped with a contact detector and tactile sensors. Experimental results were presented for a door-handle localisation application. The sequence was initialised with a vision-based estimation of the door which was then moved to increase the uncertainty to be faced by tactile sensing. The time spent for the decision about *where-to-sense-next* is significant, on the order of tens of seconds.

Furthermore, Javdani *et al.* [33] compared the information-based approach followed by Hebert with a hypothesis-pruning technique through a series of simulations. In 2014, Nikandrova *et al.* [52] proposed a probabilistic framework to make decision in grasping application focusing on modeling grasp stability under uncertainty. Specifically, they formalised two different approaches: in the former, the grasp which maximised the expected posterior stability was executed. In the latter, if the probability of a stable grasp was below a user-defined threshold, an info-gathering action was performed, maximising the reduction of probability entropy.

In summary, a set of decision-making frameworks to solve the problem of *where-to-sense-next* is available in literature. Covariance-based methods allow the robot to reason about uncertainty in an optimal way under Gaussian assumptions. Alternatively, POMPD methods based on probability entropy or Kullback Leibler Divergence are applicable to problems where the joint distribution over the state of the object can be multi-modal. Task-oriented metrics to quantify the progress towards application completion such as the probability of successful grasping have been proposed and are preferable over pure information-based metrics in order to make decisions about the next sensing actions. For the sake of clarity, a summary of the prior art on touch-based active sensing is reported in Table 1.1.

When Gaussian assumptions are not legitimate, the decision-making models available in literature can address tasks of limited complexity. Specifically, to achieve online feasibility for the action-selection, explicit and implicit upper bounds are set on:

- the complexity of the configuration space;
- the complexity of the considered action set;
- the decision-making horizon;

- the fineness of the geometric model of the object.

As a matter of fact, task **scalability** remains an issue, that is, these methods may not provide online-feasible solutions when the initial uncertainty exceeds such implicit boundaries. To benchmark the required execution time in prior art:

- grasping a Brita took approximately 750s in [2] with initial uncertainty [5cm 5cm 30deg], and decision-making done through POMDP with $h = 2$;
- localising a door handle took about 250s in [1] with approximately [10cm 10cm 360deg], and greedy decision making ($h = 1$).

In light of the practical industrial need of enabling the robot to calibrate its scene in a fast and reliable way, to the best of our knowledge, the definition of an action-inference model able to explicitly minimise the execution time of the whole task instead of just maximising the expected information gain is still an open problem.

Reference	**Application**	**Initial Uncertainty**	**Geometric Model**	**Inference Scheme**	**Action Selection**
De Geeter *et. al* [15]	Plate localisation with infrared sensing	2 DOFs Covariance matrix such that $diag(\Sigma) = [0.042m^2, 0.041m^2]$	Analytical	Extended Kalman Filter	Minimisation of the weighted covariance trace
Gadeyne *et. al* [21]	Cube localisation with force sensing	3 DOFs: [0.3m 0.3m 90deg]	Analytical	Histogram Filter	Not declared
Lefebvre *et. al* [42]	Cube in corner with Force Sensing	6 DOFs with rotation standard deviation of 10deg and translation standard deviation of 100mm	Analytical	Extended Kalman Filter	Action obtained from the Information matrix
Petrovskaya *et. al* [57]	Force-based object localisation	3 DOFs: [0.5m 0.5m 0.5m 360deg 360deg 360deg]	Polygonal mesh up to 100 faces	Scaling Series	Not declared
Hsiao *et. al* [32]	Tactile localisation for object grasping	3 DOFs: [0.05m 0.05m 0.05m 30deg]	Polygonal mesh. Number of vertices not declared	Histogram Filter	POMDP with time horizon two
Hebert *et. al* [28]	Tactile localisation of a door handle	Not declared	Polygonal mesh. Number of vertices not declared	Histogram Filter	Kullback-Leibler Divergence minimisation with time horizon one

Table 1.1: A summary of the prior art on touch-based active sensing.

1.5 Contributions and Results

Further to the analysis presented in the previous Section, the main contributions of this thesis to the state of the art are here summarised.

- An **experiment to test human approach to touch-based localisation tasks** was carried out. Facing a robot-like localisation scenario in a semi-structured environment, free to choose their action-selection strategy, 30 subjects completed the task following common patterns such as adopting the same sequence of actions and decoupling the problem into a series of subtasks. The test is detailed in Chapter 2. The results of the experiment inspired the definition of the so-called DOF Decoupling methodology which is introduced and analysed in the following Chapters.
- Chapter 3 studies examples from the state of the art on active localisation to elaborate a **model-level description of Active Sensing Tasks** in terms of configuration space, information space, action space, inference scheme and action-selection scheme. In general, each of these items influence the complexity of the active-sensing task. Concrete examples conforming to the model are provided from the literature.
- Building upon the results obtained in the human experiment presented in Chapter 2, we present the **DOF Decoupling Methodology for Active Sensing** which allows the robot programmer to formalise the active-sensing task in a scalable, DOF-decoupled fashion. Specifically, a whole localisation application can be described as a series of active sensing subtasks, each defined over a subset of the initial configuration space. Hence, the complexity of the configuration space can be minimised, thus reducing the computational effort required by the entire task.
- The **scalability** of the DOF Decoupling Methodology is validated through a localisation application with high initial uncertainty. Specifically, a solid rectangle must be localisied on top of a table. The running time to complete the task with and without decoupling is compared to highlight the gain in terms of complexity reduction in the former case. These results are presented in Section 4.2.
- **Items of increasing complexity** up to industrial relevance are localised with a Staubli RX90 robot, validating the measurement and inference model with real-life context, sensor and objects. The results of the experiment in terms of localisation reliability are presented in Section 4.3.
- **An action-selection algorithm accounting for task-related resources** such as the overall execution time is formalised and implemented

on case-study scenarios. Previous works studied the decision-making problem focusing on maximising the information gain of future sensing actions regardless of the time required to compute it. In addition, other works focused on information processing obtained through random yet quick-to-compute sensing actions. The proposed algorithm aims at filling the gap between these two approaches making the trade off between information gain and execution time explicit. Named *act-reason*, the algorithm is formalised in Section 3.4 and concrete implementations are presented in Section 4.5.

Chapter 2

Human Approach to Touch-based Object Localisation [1]

Driven by the need to analyse the decision-making process occurring during localisation tasks, a test was carried out to observe how humans approach this problem. This chapter details the motivation that led to the definition and design of the experiment, presents the set up and the protocol of the test and illustrates the observed results. The major outcome of the experiment is that humans followed the common pattern of performing the same sequence of actions decoupling the task into lower-complexity problems.

2.1 Motivation

The aim of the experiment was to observe human beings approaching an active-sensing task using a force sensor. The test was designed to replicate the uncertainty and measurement system occurring on an anthropomorphic robot arm featuring a spherical end-effector with a wrist force sensor. In particular,

[1] A summarised version of this chapter has been published in the Proceedings of the 2013 International Conference on Advanced Robotics (ICAR): Niccoló Tosi, Olivier David and Herman Bruyninckx; DOF-Decoupled Active Force Sensing (D-DAFS): A human-inspired approach to touch-based localisation tasks., 2013 16th International Conference on Advanced Robotics, 2013. p. 1-8.

the test focused on understanding whether common patterns could be identified among subjects' behaviour.

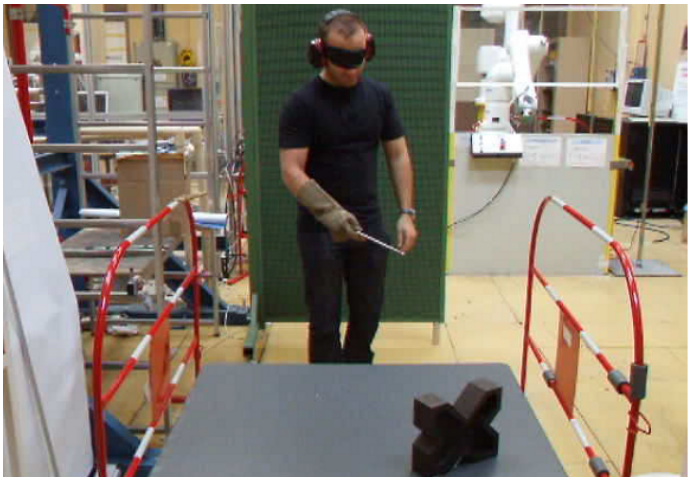

Figure 2.1: A subject undertaking the touch-based object localisation experiment. An eye cover, thick gloves and a head set were used to shut down all forms of sensing but wrist force sensing.

2.2 Experiment Set Up

The test was carried out at CEA LIST and was taken by 30 adult subjects. Throughout this chapter, the DOFs ($\{x, y, z, \theta_x, \theta_y, \theta_z\}$) used to describe the pose of the objects present in the scene are defined with respect to the reference frame depicted in Figure 2.2. During the test, the configuration space was comprised of the following items:

- two protections *p1* and *p2* assumed to be parallel and always at the same distance: $\{x, y, z, \theta_x, \theta_y, \theta_z\}_p$
- a square table: $T_{\{ABCD\}}$ with side of 1m: $\{x, y, z, \theta_x, \theta_y, \theta_z\}_T$
- the v-block object (obj_{VB}) shown in Figure 2.3: $\{x, y, z, \theta_x, \theta_y, \theta_z\}_{\text{VB}}$.

Their pose was approximately that depicted in Figure 2.1 and in Figure 2.2, even though no calibration was performed in order to keep a significant level of geometric uncertainty. With gravity preventing objects from floating

and under the assumption that the protections could only be rotated and not translated, the initial 18 DOFs were restricted to the following: $\{\{\theta_z\}_p, \{x, y, \theta_z\}_T, \{x, y, z, \theta_z\}_{\text{VB}}\}$. It is important to note that the reference convention is here reported for description purposes and it was not revealed to the subjects in order not to influence the test.

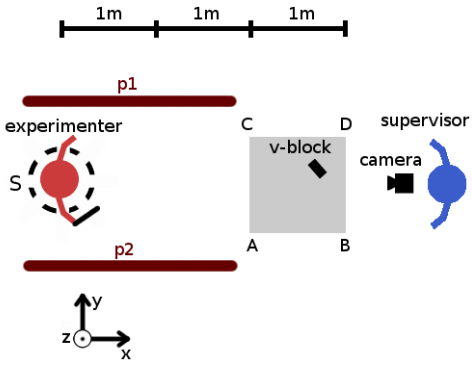

Figure 2.2: Experiment set up: the objective is to localise the v-block located on top of the table.

Figure 2.3: The v-block (obj_{VB}) used in the experiment.

2.3 Experiment Protocol

As the test started, the subjects had 30s to accurately observe all the elements composing the scene before wearing a thick glove, an eye cover and a headset.

This way, they could hold a 3D representation of the environment while performing the blind localisation. The experiment was recorded using a camera positioned beyond the table with respect to the tester. Particular attention was paid to prevent each subject from knowing the scene before her/his trial started. It was also forbidden to attend other people's trials.

Two trials were carried out to let the subjects face two different uncertainty situations, referred to as Test 1 and Test 2.

- **Test 1 - localisation of the v-block on the table:**

  Subjects were asked to position themselves on the circle S, spin once, and then localise the v-block with the information that it was located on the table. Position and orientation of the object were unknown. The task was considered achieved when the stick was steadily positioned in a horizontal position inside the v-block vertical valley, as illustrated in Figure 2.4. If the subject did not manage to complete the operation after 120s from the first contact with the object, the task was considered not accomplished.

- **Test 2 - localisation of the v-block on the table edge:**

  Subjects were asked to position themselves on the circle S, spin once, and then localise the v-block with the information that it was located on the AB edge and oriented with the V-valley parallel to AB edge. The task was considered achieved when the stick was steadily positioned in a horizontal position inside the v-block vertical valley. If the subject did not manage to complete the operation after 120s from the first contact with the object, the task was considered not accomplished.

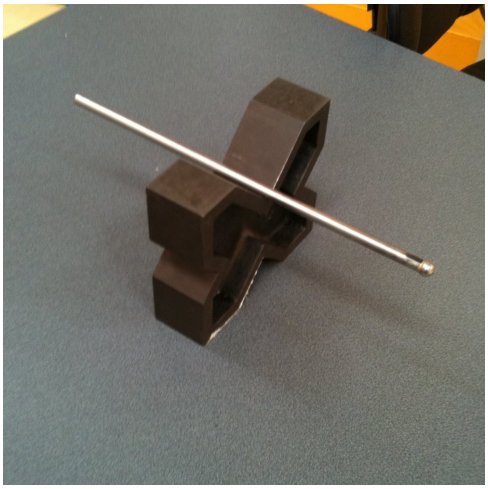

Figure 2.4: End condition: the localisation was considered over when the subject released the stick on top of the v-block.

2.4 Experiment Results

From the experiment observation, the level of initial uncertainty experienced by the subjects, due to the senses cutting-off and the spinning, changed of intensity from person to person. We consider this to depend on: self-orientation skills, fear of darkness and body equilibrium.

While performing the active-sensing task, the subjects followed common behavior patterns which can be encoded as a sequential execution of the same set of actions. For each step of the sequence, Table 2.1 presents the identified action and its correlation with the system DOFs expressed with respect to the fixed frame in Figure 2.2. For both tests, the number of subjects who decided to make the action is reported. In Figure 2.5 and 2.6, the active sensing task is represented with a graph formalism where each node corresponds to a step and reports the number of subjects who went through it. Each oriented arc corresponds to the transition between two steps, reporting the number of subject who performed it. For the sake of clarity, Figure 2.7 shows a series of snapshots of two of the 30 subjects during each of the six identified steps.

Step	Action	DOF(s)	Test 1	Test 2
1	touch *p1* or *p2*	$\{\theta_z\}_p$	6/30	4/30
2	touch AC	$\{x, \theta_z\}_T$	26/30	26/30
3	touch AB or CD	$\{y\}_T$	17/30	23/30
4	table-compliant	$\{z\}_{\text{VB}}$	23/30	21/30
5	T exploration	$\{x, y\}_{\text{VB}}$	30/30	30/30
6	obj_{VB} exploration	$\{\theta_z\}_{\text{VB}}$	30/30	30/30

Table 2.1: Correlation between scene's DOFs and actions. The table reports the sequence of steps and the associated actions undertaken by the subjects in Test 1 and Test 2.

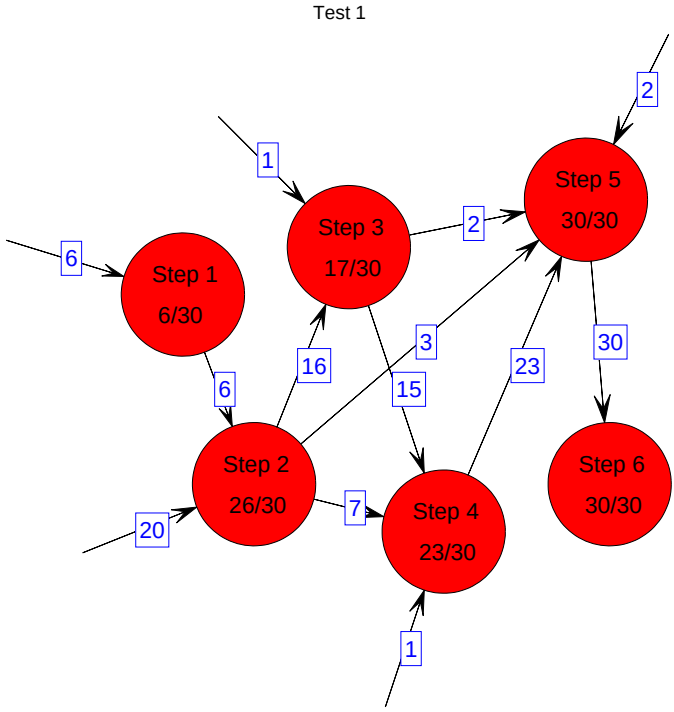

Figure 2.5: The active-sensing task in Test 1 represented as a graph. Each node corresponds to a step of the sequence, and it reports the number of subjects who undertook it, including those who began the localisation in that node. Each arrow (arc) links two steps which were executed in sequence and reports the number of subjects who performed the transition.

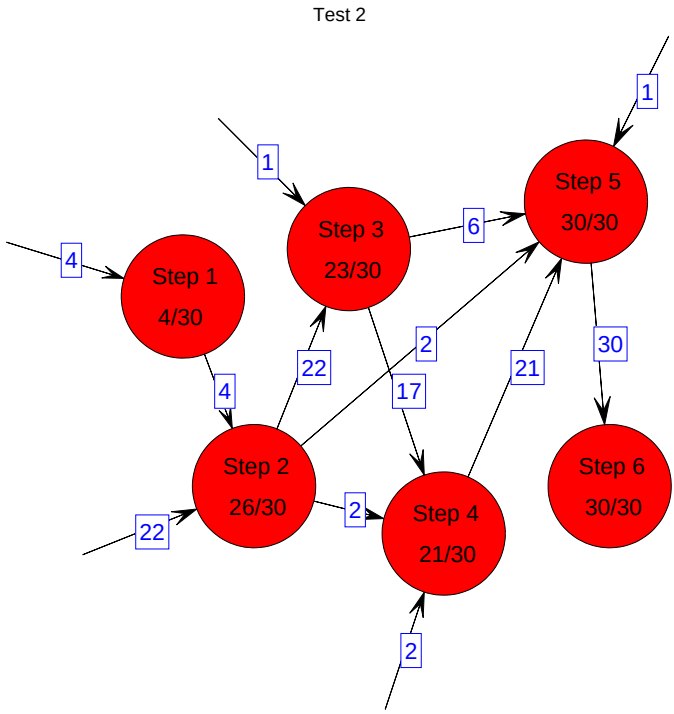

Figure 2.6: The active-sensing task in Test 2 represented as a graph. Each node corresponds to a step of the sequence, and it reports the number of subjects who undertook it, including those who began the localisation in that node. Each arrow (arc) links two steps which were executed in sequence and reports the number of subjects who performed the transition.

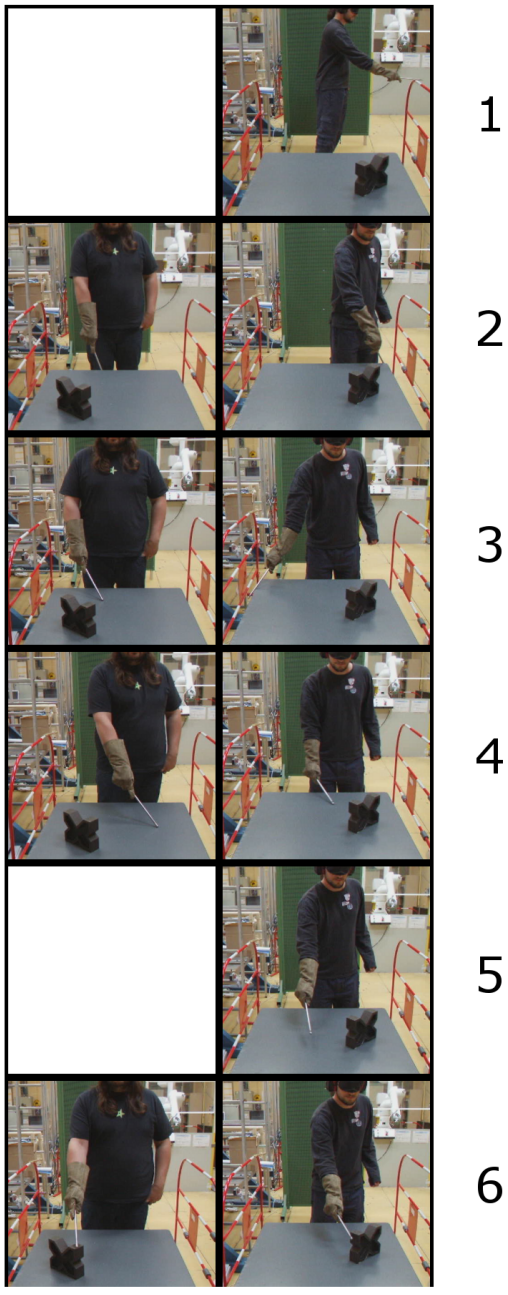

Figure 2.7: Motion sequence comparison between two subjects during the v-block localisation. The snapshots correspond to the six steps described in Table 2.1. The tester on the right hand side followed all the steps, whereas the one on the left hand side skipped step 1 and 5.

Although not all of the testers followed every step of the series, the sequence never changed. In other words, a few experimenters skipped some steps, but always followed the same action order. We consider this to depend on their prior knowledge and confidence about the specific DOFs in each step. Since all the subjects were given the same information and freely chose which strategy to follow, we deduce that the link between the uncertainty and the sequence of actions is to be found in the constraint relationships between scene objects, possibly enhanced by the rather Cartesian setup. More specifically, the knowledge of the v-block being on the table was interpreted by the subjects who decided to approach the problem by:

1. localising themselves with respect to the protections (step 1)
2. localising the table (step 2-3)
3. localising the v-block on the table (steps 4-6).

Below, the actions corresponding to each step are analysed, with particular focus on step 5 and 6 which were performed by all the subjects.

- In **step 1**, subjects extended their arms with the stick oriented downward. After contacting the protection, they slided over it.
- In **step 2**, subjects kept the stick approximately oriented downward. After contacting the AB side, they slided over it.
- In **step 3**, subjects slided the stick along the AB or CD side of the table.
- In **step 4**, subjects kept the tip of the stick in contact with the table surface.
- **Step 5** was aimed at reducing the uncertainty over the translational DOFs of the v-block by establishing a first contact with it. Three alternative strategies were chosen to perform step 5, as shown in Figure 2.8: **X-oriented**, **Y-oriented** and **Spiral** exploration, *e.g.* taking depth samples either approximately along the x or y axis, or moving the stick along an approximated spiral trajectory. Table 2.2 reports the statistics about which strategies were followed by subjects in step 5 during Test 1 and Test 2.

	X-oriented	**Y-oriented**	**Spiral**
Test 1	8/30	19/30	3/30
Test 2	26/30	4/30	0/30

Table 2.2: Exploration action options during step 5.

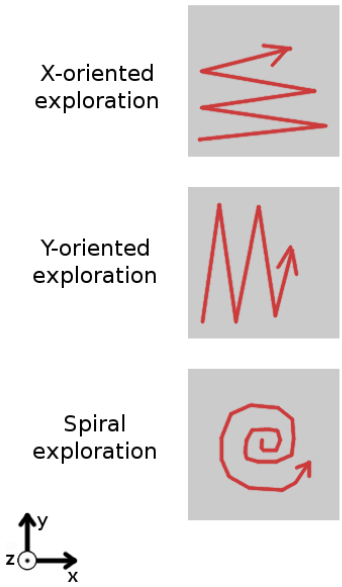

Figure 2.8: Table exploration strategies observed during step 5 (see Table 2.1): X-oriented, Y-oriented and spiral

In Test 1, the majority of the subjects chose to follow a Y-oriented exploration of the table. Since the uncertainty was homogeneous on the X and Y axes, Y-oriented exploration allowed to sense a large part of the table at each sweep, using the full length of the stick to amplify the arm movement. In test 2, the uncertainty on the position of the v-block was not homogeneous over the table and was mainly concentrated along the X axis. Since experimenters had to sense mainly the AB edge parallel to the X axis, X-oriented exploration starting from corner A was chosen by the large majority of subjects. Such a search strategy allowed to finely sense the area of high probability.

- **Step 6** represented the final phase of the localisation procedure, aimed to reduce the uncertainty over the v-block orientation. With respect to step 5, step 6 was approached by following a larger variety of strategies. Observing the videos, it was possible to identify two of them which are here analysed: touching the lateral valley of the v-block (see Figure 2.9), and touching the front face of the v-block with the stick parallel to its longer side, as shown in Figure 2.10. The two strategies are mutually independent, *i.e.* every subject

could do both of them, only one or none. Table 2.3 presents the statistics of the final-step action chosen by the subjects.

	Lateral valley	**Front face**	**Other strategies**
Test 1	20/30	3/30	7/30
Test 2	11/30	3/30	16/30

Table 2.3: Exploration actions for v-block exploration during step 6.

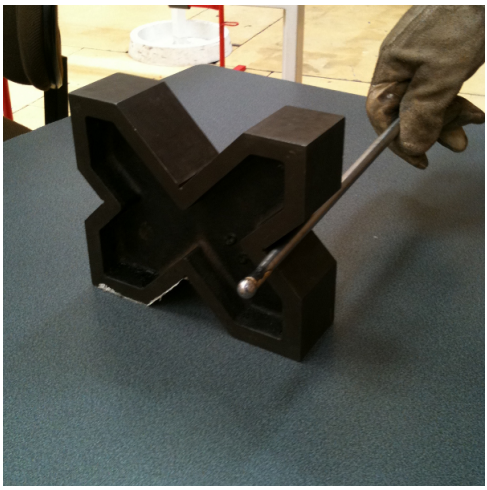

Figure 2.9: Touch on the lateral valley during step 6.

Data show how 20/30 subjects chose to touch the lateral valley in Test 1, but only 11/30 did it in Test 2. Since Test 1 presented a higher level of uncertainty over $\{\theta_z\}_{\mathrm{VB}}$, we can infer that such an action is focused on reducing the uncertainty on that specific DOF. Moreover, the lateral valley represents a stable point of interest to look for. Having reached the valley, the stick could be held inside it allowing for orientation fine-adjusting.
Touching the front face, instead, was chosen by 10% during both tests. Even though such action provides rich information on $\{\theta_z\}_{\mathrm{VB}}$, we suppose that it was considered too expensive by the subjects and therefore not convenient for the majority of the testers.

2.5 Conclusion

Having examined the results of the experiment, the following conclusions can be made:

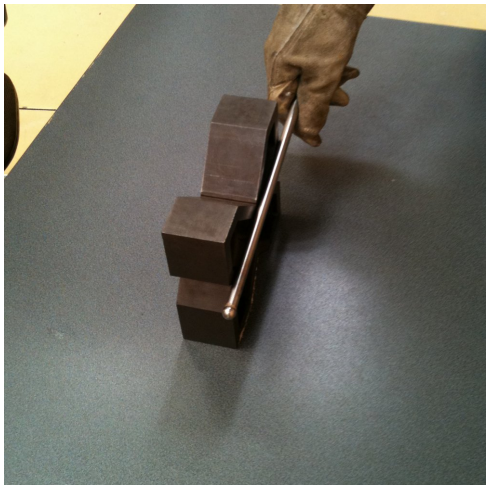

Figure 2.10: Touch on the front face during step 6.

- Human beings approached the proposed localisation task by dividing it into a **sequence of steps**. Initial uncertainty involving different DOFs resulted in different selected actions. Therefore, a direct correlation can be inferred between the actions performed by the subjects in each step and the tackled DOFs.
- **None of the subjects changed the action sequence**. Hence, we can hypothesise this to be the results of the reasoning about the information and common knowledge shared by the subjects before starting the localisation. Sensibly, one may suppose this prior information encoded in the **topological constraints** (*e.g.* the v-block is *on* the table, the table is *within* the protections) acquired by the subjects when observing the scene before undertaking the test. Further experiments with different topological relations are foreseen to verify this hypothesis.
- Subjects preferred to look for **geometric features** that guaranteed good **stability properties**, *e.g.* touching the v-block lateral valleys instead of the front face, so that it was easier to explore them with the stick. Moreover, when exploring the lateral valleys, the stick was aligned as in the target configuration. Hence, features explorable with the end effector oriented as in the target configuration seem to be preferable over features that will require more complex transitions afterwards.
- The **actions** chosen by the subjects seemed to be **influenced by the dynamic properties of the objects**, *e.g.* their weight and friction

coefficients. Though without proof yet, we expect a different sequence of actions to be selected in the presence of different properties, *e.g.* lighter (movable) objects. Further experiments could be set up to verify this hypothesis, for instance asking subjects to localise the same object with and without table fixture.

2.6 Discussion

This Section critically discusses the hypotheses and the choices made in the experiment described in this Chapter and suggests possible enhancements to extend the obtained results.

1. The experiment set up was comprised of mostly regular objects whose relative orientation was either approximately parallel or orthogonal. To some extent, such topological constraints may have influenced the behaviour of the subjects. Experiments with a less structured arrangements of the objects are foreseen to investigate the effect of a less Cartesian environment on the decision making. For example, one could carry out the same test with table side AB significantly mis-aligned with respect to the protections $p1$ and $p2$.

2. The pose of the objects was described using a frame of reference attached to the world. For the adopted set up, this allowed us to observe a correlation between actions and DOFs, since most of the frames were translated and not rotated. In a more general case, one has to describe the reference frames with respect to different features, for instance the pose of the v-block as a transformation expressed with respect to the reference frame of the table.

3. Provided with 30s to observe the scene, all the subjects were assumed to share the same information about the environment. Further research may quantify the impact of visual knowledge by comparing the results between subjects who have seen the scene and those who have only seen a map of it as in 2.2.

Chapter 3

DOF Decoupling

In this Chapter, we introduce a **model** to make the concept of **active-sensing task** concrete using five modelling primitives: configuration space, information space, action space, inference scheme and action-selection scheme. The presented model is suitable for localisation applications which are not restricted to touch-based sensors only. Previous object-localisation works conform to the model as application-specific instances, and concrete examples from the literature are provided.
The **DOF Decoupling Methodology** is presented as a formalism to describe localisation application as a sequence of active-sensing tasks each of them defined on the most appropriate configuration space. The name is inspired by the outcome of Chapter 2, where humans approached a localisation task by changing DOFs of the configuration space to reduce the complexity of the active-sensing task at run time.
In addition, this Chapter introduces a new action-selection **algorithm** named *act-reason* which allows the robot to adjust the complexity of the action space at run time, explicitly trading off information gain with planning and execution cost.

3.1 Motivation

In Bayesian touch-based localisation applications, the pose of an object is estimated by updating a belief over the configuration space, conditioned on the obtained measurements. Typically, such probability distribution becomes multi-modal, due to the local nature of the available information, *e.g.* taking the form of a contact point and a normal vector. Under these conditions, optimal and computationally-efficient parametric Bayesian filtering models such as the Kalman Filter are not suited to capture the different modes of the probability

distribution. Hence, non-parametric inference models relying on a discretisation of the probability distribution must be adopted, such as grid-based methods [28] [30] or Particle Filter [22] [57]. Unfortunately, the computational complexity of these inference schemes linearly increases with that of the configuration space. As a matter of fact, this sets an upper bound on the size of the problems that can be treated online since the available resources are always finite. For instance, a grid-based inference model for a 6DOF localisation task with initial uncertainty of [1m 1m 1m 360deg 360deg 360deg], with resolution of 5mm in translation and 2 deg in rotation, corresponds to approximately 3×10^{12} nodes: this may represent a significant requirement of computational and memory resources at run time. Moreover, making decisions about where-to-sense-next reasoning in the belief state is computationally cumbersome, as the outcome of each considered sensing action requires a simulation and a posterior update. So, the complexity is exponential in the number of DOFs of the configuration space and in the time horizon.

The experiment presented in Chapter 2 has shown a natural inclination of human beings to decouple a localisation application into a **sequence of lower-complexity problems**, exploiting the knowledge about the environment and the relations between the objects present in the scene. Building upon this result, our intuition is that also robots should be able to divide a localisation application into a series of subtask adapting the problem complexity at run time. This would allow the robot **to improve the scalability** of the active sensing, thus becoming capable of tackling tasks of **higher complexity**. Backed by the results of this test, we formulate DOF Decoupling as a general framework to describe a robot-localisation application in a modular fashion to minimise the complexity of the configuration space.

The formulation proposed here is meant to be a model-level generalisation of active-sensing task description, and previous works of touch-based object localisation conform to this framework as concrete implementations of the model. Section 3.2 revises active sensing examples from the literature which can be described in terms of configuration space, information space, action space, inference scheme and action-selection scheme. The presented model goes under the name of Active Sensing Task model. In Section 3.3, the DOF Decoupling Methodology is introduced, which allows the robot to decouple a whole localisation application into a sequence of active-sensing subtasks defined on different configuration spaces. Section 3.4 presents an action-selection algorithm which allows the robot to deterministically allocate the time spent for evaluating *and* executing the next sensing action, hence explicitly setting the cardinality of the action space at run time.

3.2 Active Sensing Task Model

Prior active-sensing tasks in literature can be described by a set of instances of the following primitive concepts:

- **Configuration** space X, *i.e.* the model to describe where the objects are in the world with respect to the robot reference frame.
- **Information** space $P(X_t|a_{0:t}, z_{0:t})$, *i.e.* the probability distribution that encodes the knowledge about the configuration space at time t, given the sequence of actions $a_{0:t}$ and observations $z_{0:t}$.
- **Action** space A, *i.e.* the set of actions the robot can execute. This depends on the motion capabilities and the sensory apparatus of the robot.
- **Inference** scheme, *i.e.* the way of computing $P(X_t|a_{0:t}, z_{0:t})$.
- **Action-selection** scheme, *i.e.* the algorithm to make decisions and choose one of the items in the action space.

The **descending order** in which the primitive concepts are itemised encodes a correlation in terms of **complexity** of the active sensing task. Specifically, the complexity of the configuration space influences that of all the other primitives. The information space complexity influences that of action, inference and action selection, and so on and so forth. The aforementioned primitive concepts represent the prime elements to describe an active sensing task. In the following subsections, examples from the literature are presented as concrete instances of the Active Sensing Task Model.

3.2.1 Localisation with Infrared Sensing

In [14], De Geeter *et al.* localised a plate in 3DOFs using infrared sensing to detect its edges. The target was a pre-defined point on the plate. The initial uncertainty on its x-y position was encoded as a covariance matrix with $diag\{0.041m^2,\ 0.042m^2\}$.

- *Configuration space*: 3DOF pose of the plate.
- *Information space*: Gaussian posterior distribution over the pose of the plate.
- *Action space*: infrared probing to detect the edges of the plate.
- *Inference scheme*: Extended Kalman Filter.

- *Action-selection scheme*: actions selected as to minimise the weighted trace of the covariance matrix.

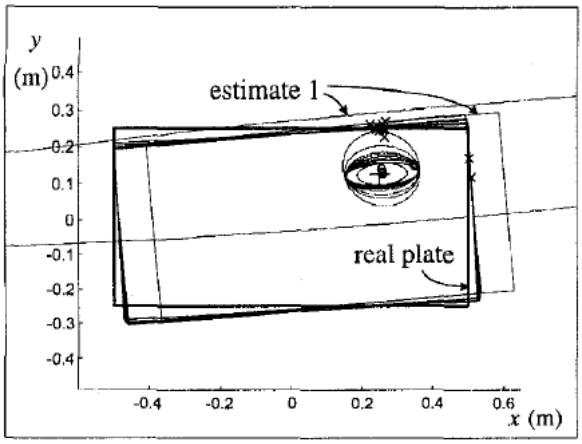

Figure 3.1: Plate localisation using infrared sensing. Figure as in [14].

3.2.2 Localisation with Force Sensing (ex. 1)

In [21], Gadeyne and Bruyninckx localised a cube in 3DOFs using force sensing with initial uncertainty of [0.3m 0.3m 90deg].

- *Configuration space*: 3DOF pose of the cube with respect to the robot frame of reference.
- *Information space*: uniform grid discretising the posterior distribution over the configuration space.
- *Action space*: force-based touch sensing resulting in contact point and normal vectors of the touched surface.
- *Inference scheme*: Histogram filter.
- *Action-selection scheme*: not declared.

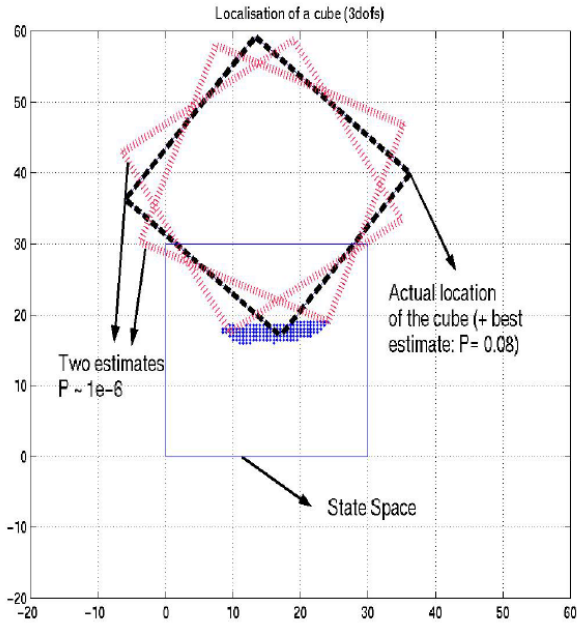

Figure 3.2: Cube localisation in 3DOFs using force sensing. The square indicates the possible x-y coordinates of the bottom vertex of the cube. Image from [21].

3.2.3 Localisation with Force Sensing (ex. 2)

In [56], Petrovskaya *et al.* localised a set of objects modelled with a polygonal mesh using force sensing with initial uncertainty up to [0.5m 0.5m 0.5m 360deg 360deg 360deg].

- *Configuration space*: 6DOF pose of the object with respect to the robot frame of reference.
- *Information space*: sampled posterior distribution over the configuration space.
- *Action space*: depth-sampling motions with a spherical end effector coupled with a force-torque sensor. Collected measurements comprised of contact point and normal vector.
- *Inference scheme*: Scaling Series.
- *Action-selection scheme*: random selection of pre-determined actions.

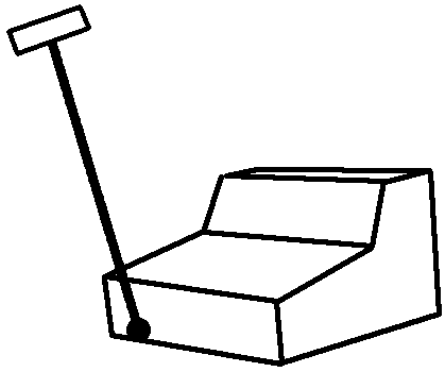

Figure 3.3: Object Localisation in 6D with force sensing. Image inspired by [56].

3.2.4 Localisation and Grasping with Tactile Sensing

For grasping applications, in [30], Hsiao localised a set of objects modelled as a polygonal mesh, with initial uncertainty up to [0.05m 0.05m 30deg].

- *Configuration space*: 3DOF pose of the object with respect to the robot frame of reference.
- *Information space*: uniform grid discretising the posterior distribution over the configuration space.
- *Action space*: the set of action primitives was defined off-line with respect to the object geometry. It was comprised of hand trajectories executed with a contact detector on the palm and tactile sensors on the fingers.
- *Inference scheme*: Histogram filter.
- *Action-selection scheme*: POMDP with time horizon $h = 2$, *i.e.* actions were chosen as to maximise the expected reward after two actions. The probability of successful grasp was used as progress metric.

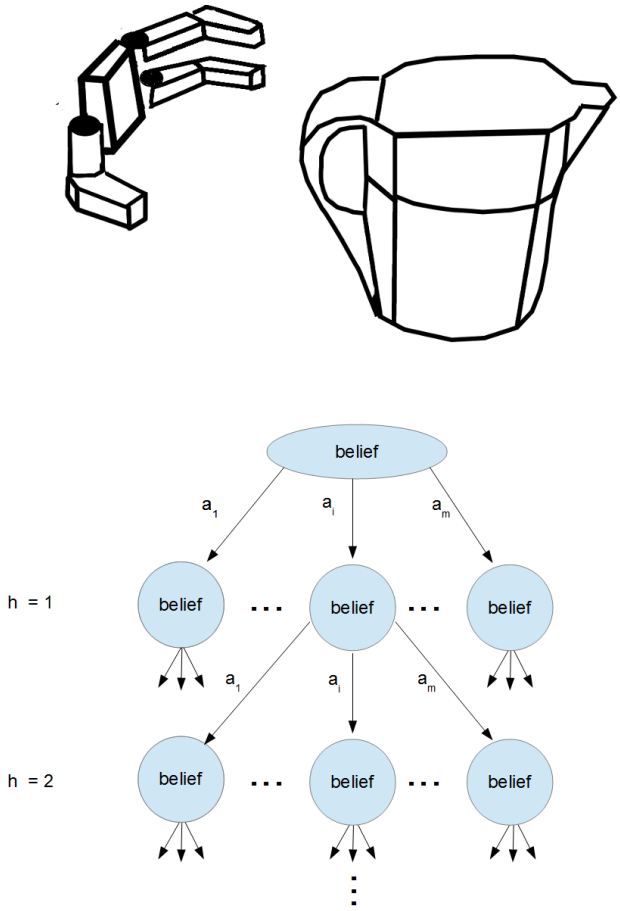

Figure 3.4: 3DOF Object localisation with Barret hand equipped with tactile sensors. initial uncertainty: [0.05m 0.05m 30deg]. Actions selected through POMDP with decision-making horizon 2. Image inspired by [32].

3.3 DOF Decoupling Methodology

The curse of dimensionality described in Section 3.1 is a strong driver to reduce the complexity of localisation applications. In the experiment

presented in Chapter 2, human subjects have shown the common behavioural pattern of decoupling a localisation application into a **sequence of lower-complexity problems**. Inspired by these findings and focusing on reducing the computational complexity of localisation applications, we here introduce the DOF Decoupling Methodology. Under this formalism, a localisation application is a composite task (see Def. 1.2) composed by a **sequence of subtasks**. Each subtask is defined as an active sensing task (see Section 3.2) with a configuration space **of lower complexity**. Both referring to an active-sensing task, the terms "subtask" and "localisation application" are here adopted to highlight that the configuration space of the former is a subset of that of the latter. Consequently, the run-time action-selection complexity of a subtask is reduced, which often represents the computational bottle neck of localisation tasks.
Building upon [43], under the DOF Decoupling formalism, a localisation application is represented as an oriented graph named *task graph*.

Def. 3.1. Task Graph: *oriented graph representing a localisation application in which each node corresponds to an active-sensing task defined in its specific configuration space.*

The task graph formalism allows us to encode the sequential order in which subtasks can be executed. Arcs encode transitions between subtasks which are triggered by events, for instance: a sensor signal reaching some specified threshold, a change in the level of uncertainty, a human-specified occurrence. In this context, we define *policy* as follows:

Def. 3.2. Policy: *a path through subtasks which are connected by transitions.*

Different policies may be defined in a task graph. For instance, this could be done further to an empirical observation, as in the case study described in Section 4.2. In order to select a policy, graph-search techniques from the literature can be applied, *e.g.* random selection, breadth-first or depth-first. Hence, as proposed in [43], an active sensing localisation task requires decision making at two different levels:

1. to select a policy, *i.e.* the sequence of subtasks to execute;
2. to select action within each subtask, as described in Section 3.2.

In general, the structure of the task graph is application-dependent. For the sake of clarity, Figure 3.5 illustrates an example of a localisation application described as a task graph with subtasks defined over three different configuration spaces: C1, C2 and C3. Three policies are available in the presented case, depicted with arrows of different patterns. Subtasks are represented as a circle. It is important to note that several subtasks may be defined on the same configuration space, *e.g.* with different information space, action space, inference and action-selection scheme. Transition events are depicted with a filled circle.

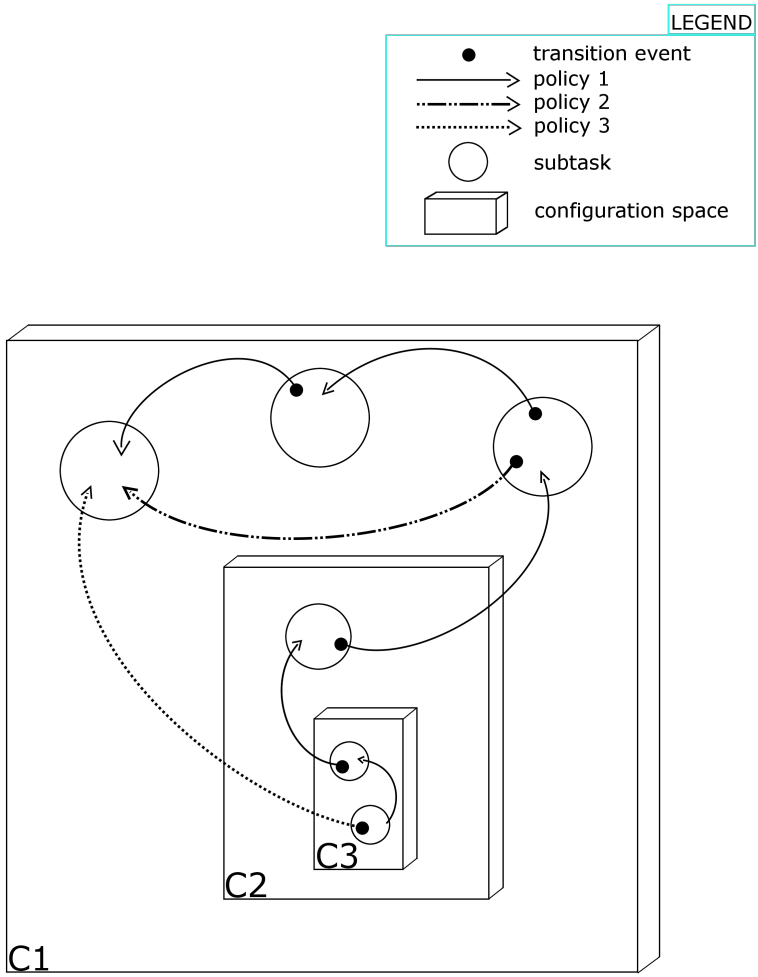

Figure 3.5: Example of *task graph* as defined in the DOF Decoupling Methodology. Graphically, solid rectangles represent configuration spaces, whereas circles represent tasks. In this specific case, the task is defined over three configuration spaces: C1, C2 and C3, presenting three, one and two tasks respectively. The localisation is initialised in configuration space C3. Three policies are depicted with arrows of different patterns, each of them representing a possible way through the tasks defined over C1, C2 and C3.

In the following subsections, two examples of DOF Decoupling from the literature are presented as concrete instances of the methodology.

3.3.1 Cube-in-Corner with Contact-Formation Graph

A first example of task decoupling for touch-based localisation is presented in [43], where a cube-in-corner active-sensing application is performed using force sensing. The configuration space is composed by twelve DOFs representing the pose of the table e with respect to the fixed frame w $\{x, y, z, \theta_x, \theta_y, \theta_z\}_e$ and the pose of the gripper g with respect to the manipulated object m $\{x, y, z, \theta_x, \theta_y, \theta_z\}_g$, as depicted in Figure 3.6.

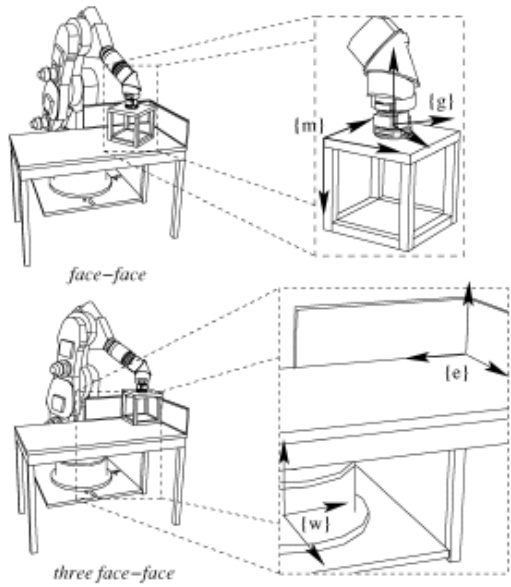

Figure 3.6: Frames of reference used in [43].

The knowledge about the possible discrete contact configurations between the cube and the table is encoded in the so-called contact-formation graph. In a DOF Decoupling framework, this corresponds to a task graph created from a subset of the potentially enormous number of discrete configurations between the cube and the table, obtained from the disassembly of the two parts [41]. More explicitly, each contact formation corresponds to an active sensing subtask, described as in subsection 3.2. As depicted in Figure 3.7 using the same graphical conventions as in Figure 3.5, each node represents a contact formation (*e.g.* vertex-vertex, face-vertex) and the arcs correspond to compliant transitions

between them. Given the cubic shape of m, modelling the uncertainty on the DOFs as a multivariate Gaussian and linearising the measurement equations, in each subtask it was possible to:

1. calculate the expected Fisher information matrix I_f analytically
2. eigen-value decompose I_f as $I_f = U\Theta U^T$, where Θ is a diagonalised matrix with the expected information gain which can be collected on the directions expressed by the columns of U^T
3. identify the observable DOFs of each contact formation, named *spanning set*, as the union of the eigenvectors in U^T which correspond to non-zero eigenvalues in Θ.

In practice, the actual decoupling was applied at action-selection level, with actions selected as to excite a linear combination of a subset of the DOFs of the configuration space, corresponding to the columns of the U^T matrix (for instance, Figure 3.8 depicts a sensing action during a vertex-face subtask). More precisely, the configuration space C1 was always the same (with 12DOFs), whereas different action-selection strategies were applied depending on the current contact formation and its associated observable DOFs. Hence, each node of the *task graph* in Figure 3.7 is an instance of the model presented in Section 3.2 as detailed below:

- *Configuration space*: 12 DOFs to represent configuration space of the table and the gripper pose
- *Information space*: multivariate Gaussian distribution over the configuration space
- *Action space*: compliant motion primitives
- *Inference scheme*: Kalman Filter
- *Action-selection scheme*: direction of motion selected as to excite the contact-formation *spanning set*.

The selection of a *policy* through subtasks was formulated as a constrained optimisation of the discrete sequence minimising the number of transitions or the total execution time, guaranteeing the observation of all the DOFs in the configuration space.

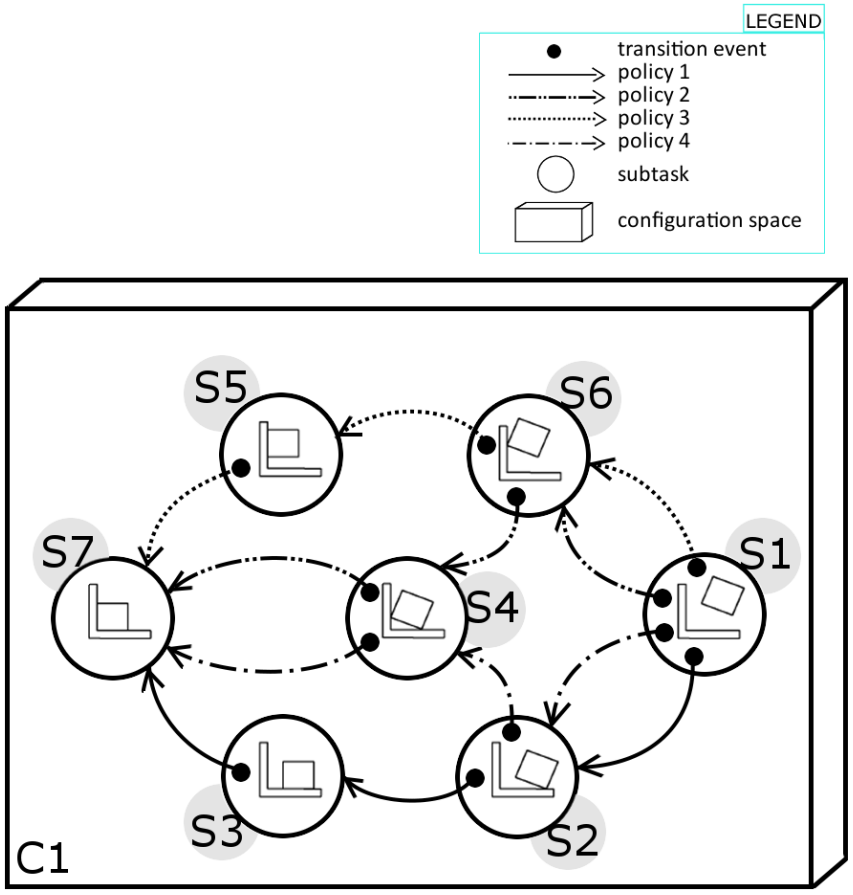

Figure 3.7: Example of *task graph* for a cube in corner application generated from the assembly of the two objects. Encoded as a node, each task correponds to a discrete contact formation between the cube and the edged table. The same graphical notation as in Figure 3.5 is used.

For the sake of clarity, the face-face subtask S3 is here described under the DOF Decoupling Methology. The reported *spanning set* is taken as-is from [43].

- *Configuration space*: 12DOF configuration space of the table and the gripper pose $\left\{\{x, y, z, \theta_x, \theta_y, \theta_z\}_e, \{x, y, z, \theta_x, \theta_y, \theta_z\}_g\right\}$
- *Information space*: multivariate Gaussian distribution over the configuration space
- *Action space*: face-face compliant motions

- *Inference scheme*: Kalman Filter
- *Action-selection scheme*: actions chosen as to excite the following linear combination of DOFs: $\big\{\{\theta_x, \theta_y\}_e, \{\theta_x, \theta_y\}_g, \{z\}_e + \{z\}_g\big\}$.

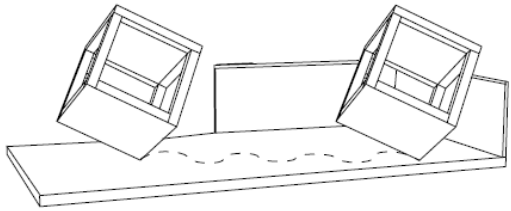

Figure 3.8: Example of active-sensing motion within a vertex-face contact formation.

3.3.2 Door-handle Localisation with Vision and Tactile Sensing

In [28], an object localisation for grasping application is presented using a robot equipped with a three-finger hand and a camera. Specifically, the task is to identify the pose of a door handle with respect to the robot frame. This is decoupled into a first subtask in S1 where the door is registered by a vision-based sensor, and a second subtask S2 where a three-finger robot hand performs tactile exploration in order to localise the handle. The observed task-decoupling is depicted in Figure 3.9. Both S1 and S2 conform to the Active Sensing Task Model, and can be described as follows.
S1:

- *Configuration space*: 6DOF pose of the door with respect to the robot frame.
- *Information space*: not explicitly declared. Possibly, a Gaussian posterior distribution over the configuration space.
- *Action space*: camera rotations.
- *Inference scheme*: not declared.
- *Action-selection scheme*: not declared, possibly an exhaustive scan of the room.

S2:

- *Configuration space*: 3DOF pose of the door handle with respect to the robot frame.
- *Information space*: uniform grid discretising the posterior distribution over the handle pose.
- *Action space*: pre-defined hand trajectories with a palm contact-detector and and tactile sensors on the fingers.
- *Inference scheme*: Histogram Filter.
- *Action-selection scheme*: actions selected as to maximise the Kullback-Leibler divergence.

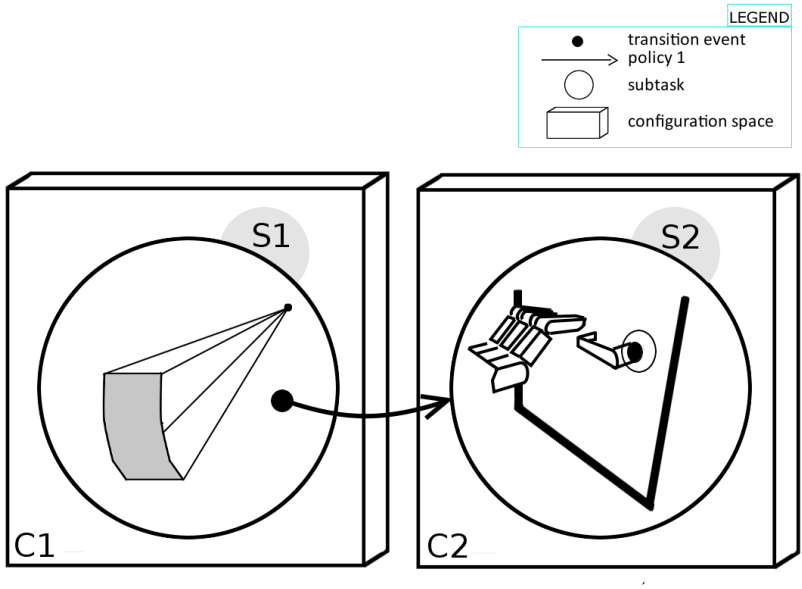

Figure 3.9: Door-handle Localisation with DOF Decoupling. The task is divided into a first vision-based localisation (S1) followed by a tactile localisation task (S2). Images inspired by [28].

3.4 Act-Reason: Making Decisions Over a Flexible Action Space [1]

In this Section, a new **algorithm** for action selection is introduced. To the best of our knowledge, previous works on active sensing made decisions about where to sense next maximising some reward function choosing from a pre-defined action space composed of a set of sensing motions either specified by a human demonstrator or automatically generated off-line [32][28][52]. This resulted in an implicit choice on the complexity of the action space, hence also of the whole active-sensing task, as described in Section 3.2. In practice, this often caused the robot to devote a significant percentage of the running time to reasoning instead of acting.
In the framework of the Active Sensing Task Model, there is the need **to make the choice on the complexity of the action space explicit**, thus allowing the robot to adjust the action-selection complexity at run time. With respect to the state of the art, we introduce a new action-selection scheme named *act-reason* which allows the robot to find a solution to the problem of where to sense next constraining the time t_{alloc} spent for the next action, including reasoning and motion execution. Therefore, the complexity of the action space depends on t_{alloc} and the time required to reason about each action. In *act-reason*, t_{alloc} is a design parameter that sets the robot behaviour from info-gathering (increasing t_{alloc}) to action-oriented (reducing t_{alloc}). In particular, this makes the action space flexible, so the complexity of the decision making can variate at run time.

3.4.1 Act-reason Algorithm

Previous related works [28][32] tackled the problem of choosing the next best action a^* by first generating a complete set of candidate actions $\{a\}$, then selecting the one that maximises some specified reward function:

$$a^* = \arg\max_a E[r(a)]. \tag{3.1}$$

Often, the reward r corresponds to the information gain [43][28] or some task-oriented metric, such as the probability of grasping success as proposed in [30]. In this Section, we present an action-selection algorithm that is irrespective of the chosen reward function, so it may apply to a generic robot task. For each sensing action a_i, the proposed scheme assumes the robot to be able to:

[1] The formalisation of the action-selection algorithm presented in this Section has been published in the Proceedings of the 2014 International Conference on Robotics and Automation (ICRA): Niccoló Tosi, Olivier David and Herman Bruyninckx; Action Selection for Touch-based Localisation Trading Off Information Gain and Execution Time, International Conference on Robotics and Automation, 2014[66].

- generate the associated motion
- evaluate the expected reward
- estimate the execution time t_{exec}
- estimate the evaluation time t_{eval}.

The *act-reason* algorithm is presented in pseudo-code in Alg. 1, with t_{alloc} being the allocated time to evaluate and execute the next action. Each time the robot needs to make a decision, the best solution a^* is initialised to "stay still", unless an open-loop action is available, *e.g.* a user-defined action (lines 2-8). Then, the timer starts (line 9). While the execution time of the current best solution is smaller than the available time ($t_{alloc} - t$.elapsed()), candidate actions are generated and evaluated, updating the best solution if the expected information is greater than the current one (lines 16-18). As the available time intercepts the estimated execution time of the current best solution, the reasoning is over and a^* is performed. The condition at line 14 ensures that the next candidate to evaluate respects the time constraint.
Figure 3.10 graphically illustrates the action-selection at different time steps: the starting time t_0, a generic instant t_i and t_{final} when a^* is executed. In the illustrated case, an open-loop action is available, so a^* is initialised to a_{OL}. As time passes, a^* is updated, until t.elapsed() intersects $t_{exec}(a^*)$, and the current best action is executed.

Algorithm 1 Action Selection with Act-Reason

1: set t_{alloc}
2: **if** a_{OL} is available **then**
3: $a^* = a_{OL}$
4: $E[r^*] = E[r(a_{OL})]$
5: **else**
6: a^* = "stay still"
7: $E[r^*] = 0$
8: **end if**
9: t.start()
10: $i = 0$
11: **while** $t_{exec}(a^*) < t_{alloc} - t$.elapsed() **do**
12: $i := i + 1$
13: a_i = generateCandidateAction()
14: **if** $t_{exec}(a_i) + t_{eval}(a_i) < t_{alloc} - t$.elapsed() **then**
15: compute $E[r(a_i)]$
16: **if** $E[r(a_i)] > E[r(a^*)]$ **then**
17: $a^* = a_i$
18: **end if**
19: **end if**
20: **end while**
21: **if** a^* == "stay still" **then**
22: $t_{alloc} := 2\,t_{alloc}$
23: go back to line 2
24: **end if**
25: execute a^*

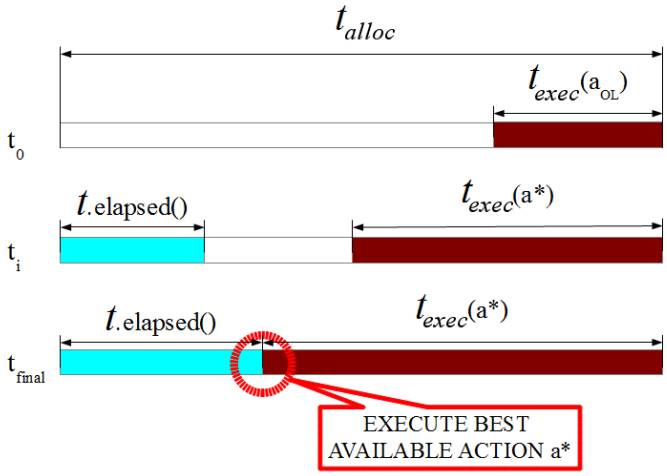

Figure 3.10: Action-selection time line with act-reason decision making. New candidate actions are generated and evaluated until the available time equals that required to execute the current best solution.

3.5 Conclusion

Generalising from previous examples in literature, a **model-level description of active sensing tasks** based on five primitive concepts has been proposed. Inspired by the results obtained in the human experiment presented in Chapter 2, the DOF Decoupling Methodology has been introduced as a new **meta-model to describe active-sensing tasks**. This allows the robot to decouple the whole task into a sequence of subtasks defined in configuration spaces of lower complexity, thus **reducing the complexity of the action-selection process** which represents the computational bottle neck of touch-based active sensing. In order to highlight the benefit of DOF Decoupling, Section 4.2 presents a high-uncertainty object localisation comparing the time required to complete the task with and without decoupling.
Act-reason has been presented as generic **action-selection algorithm** to choose where-to-sense-next making the action space flexible. Specifically, this gives the robot programmer the capability to mutate the number of actions to consider at run time, variating the allocated time to compute and execute the next one. Section 4.5 presents a robot application where *act-reason* is adopted to reduce the time required to localise a solid object adapting the action space with respect to the current uncertainty.

Chapter 4

Applications: Simulations and Robot Experiments

This chapter presents a set of applications of the DOF Decoupling Methodology to touch-based localisation. After a description of the adopted measurement apparatus installed on a Staubli RX90 depicted in Figure 4.1, examples of solid object localisation are both simulated and executed with the robot. The main objective of this chapter is to (i) show the potential of the proposed methodology in tackling high-uncertainty problems, (ii) validate the measurement model and inference scheme with concrete localisation examples of real-life objects of increasing complexity and (iii) propose task-oriented decision-making models suitable for online applications.

4.1 Measurement Model

Equipped with a spherical end-effector coupled with a force-torque sensor as shown in Figure 4.2, the robot has to localise a solid object modelled as a polygonal mesh composed of faces $\{f_i\}$ and their associated normal vectors $\{n_i\}$. Throughout this Chapter, we will refer to a single $< f_i, n_i >$ tuple as *patch*. The pose of the object is represented by the state vector $\mathbf{x}$, which collects its position and roll-pitch-yaw orientation with respect to the robot frame of reference. In a Bayesian framework, the information on the pose of the object is encoded in the joint probability distribution over $\mathbf{x}$ conditioned on the measurement z: $P(\mathbf{x}|z)$.
The robot acquires information about $\mathbf{x}$ both when moving in free space and when in contact with the object, assuming that interactions only take place with its end-effector. In this context, an action consists in the end-effector

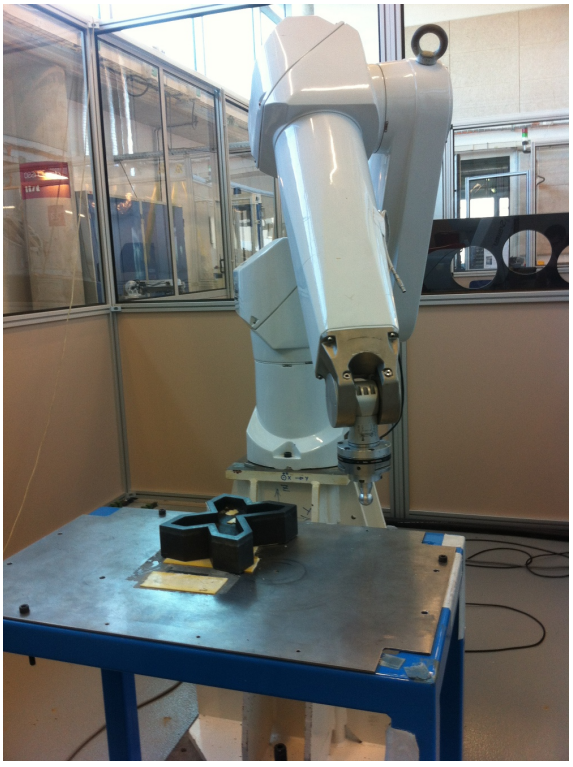

Figure 4.1: The Staubli RX90 robot used in the experiments.

following a trajectory τ, sweeping a volume $\psi(\tau)$. Concretely, τ is a collection of end-effector frames $\{\phi\}$ used as requirement for the robot controller. When a contact is detected, the motion is stopped with the end-effector in frame ϕ_c. In case of no-contact, the whole trajectory is executed with the end-effector reaching pose ϕ_{end}. The sensory apparatus is comprised of a force-torque sensor used as collision detector setting an amplitude threshold on the measured forces. Specifically, it detects whether the end-effector is in contact with the object (C) or is moving in free space (NC):

$$z_{CD} = \begin{cases} C, \text{ if the force resultant reaches the threshold} \\ NC, \text{ otherwise.} \end{cases} \tag{4.1}$$

When the end-effector touches the object, the point of contact z_P and the normal vector z_n of the touched point can be estimated following the procedure described by Kitagaki *et al.* [38], under the assumption of frictionless contact

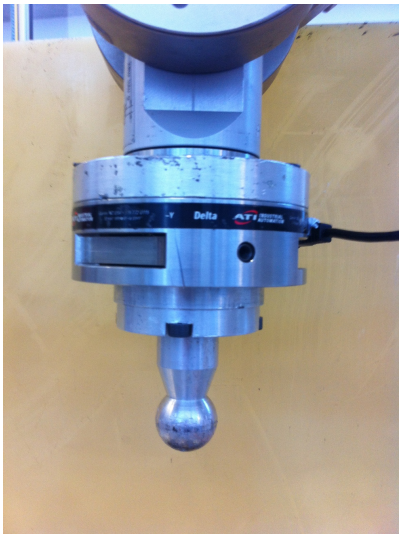

Figure 4.2: End-effector set up with coupled force-torque sensor installed on the Staubli RX90 used in the experimental test.

(further details in the Appendix). We will refer to this touch measurement as z_{touch}:

$$z_{touch} = \{z_P, z_n\}. \tag{4.2}$$

As in [31], in order to define the observation models, we consider two possible cases: (i) the robot reaches the end of the trajectory and (ii) the robot contacts the object stopping in ϕ_c.

4.1.1 The robot reaches the end of the trajectory without contact

Intuitively, we need to model the probability that the robot executes the trajectory τ without touching the object in a given pose $\mathbf{x}$. The boolean function $\xi(\mathbf{x}, \tau)$ is used to indicate whether $\psi(\tau)$ intersects the volume of the object (Eq. 4.3) :

$$\xi(\mathbf{x}, \tau) = \begin{cases} 1, \text{ if } \psi(\tau) \text{ intersects the object in pose } \mathbf{x} \\ 0, \text{ otherwise.} \end{cases} \tag{4.3}$$

Building upon [28], the contact-detection measurement is modelled as in Eq. 4.4. α and β represent the false-negative and false-positive rate for the contact detection.

$$P(z_{CD}|\mathbf{x},\tau) = \begin{cases} P(z_{CD} = NC|\xi(\mathbf{x},\tau) = 1) = \alpha \\ P(z_{CD} = C|\xi(\mathbf{x},\tau) = 1) = 1 - \alpha \\ P(z_{CD} = C|\xi(\mathbf{x},\tau) = 0) = \beta \\ P(z_{CD} = NC|\xi(\mathbf{x},\tau) = 0) = 1 - \beta. \end{cases} \tag{4.4}$$

The false-negative rate α corresponds to the probability of not measuring contact when the sweep intersects the volume occupied by the object. This coefficient captures two main sources of uncertainty. Firstly, when the robot measures $z_{CD} = NC$, it might actually be in contact with the object but in some configuration which prevents the FT signals from reaching the thresholds. Figure 4.3.a illustrates an example of such false-negative measurement situation when the limited stiffness of the controller and the low surface friction allow the robot to contact the object and slide over it without exceeding the thresholds. Secondly, α also captures the uncertainty related to the mismatch between the geometric model and the actual object which may cause the robot to intersect the polygonal mesh without measuring any real contact. An example of the latter situation is depicted in Figure 4.3.b.
The false-positive rate β encodes the probability of measuring contact when the sweep does not intersect the object volume. The threshold to detect contact is set on the resultant force and is parametrised with respect to the maximum standard deviation error among those of $\{Fx, Fy, Fz\}$. This guarantees a lower false contact rate than other thresholding policies, *e.g.* ellipsoidal or parallelepiped. In addition, we make the assumption of modelling the object with a polygonal mesh that is external or at most coincident to the nominal geometry. Moreover, the limited speed of the robot makes inertial effects minimal. Under these conditions, the probability of measuring a false contact is negligible. This corresponds to assuming $\beta \approx 0$.
In practice, the likelihood function P_{free} corresponding to a sweep without contact along τ and object pose $\mathbf{x}$ is expresed in Eq. 4.5.

$$P_{free}(\mathbf{x},\tau) = P(NC|\xi(\mathbf{x},\tau)). \tag{4.5}$$

4.1.2 The robot contacts the object

Let us name τ_c the trajectory executed by the robot following τ up to pose ϕ_c on which the contact is detected. The measurement model expresses the probability of observing no-contact along τ_c, plus a contact in ϕ_c with touch measurement z_{touch}:

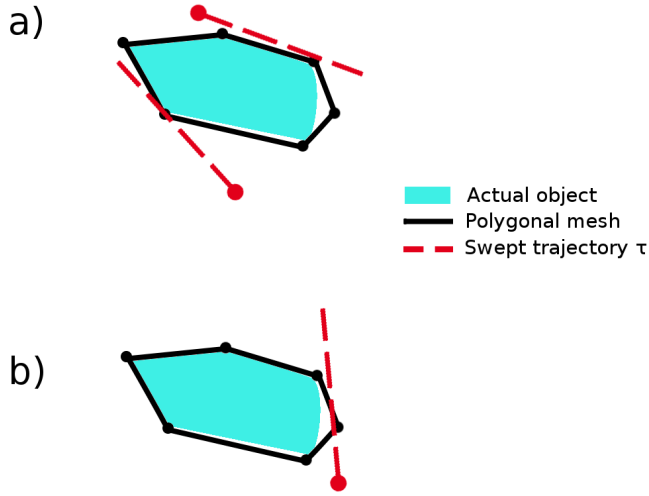

Figure 4.3: Possible false-negative contact-detection situation: a) the end effector touches the object sliding over it without exceeding the force threshold; b) the end effector sweeps through the geometric model without touching the real object.

$$P_{contact}(\mathbf{x}, \tau) = P_{free}(\mathbf{x}, \tau_c) P(z_{CD}, z_{touch} | \mathbf{x}, \phi_c). \tag{4.6}$$

$P_{free}(\mathbf{x}, \tau_c)$ can be calculated as in Eq. 4.5. By definition of conditional probability:

$$P(z_{CD}, z_{touch} | \mathbf{x}, \phi_c) = P(z_{touch} | \mathbf{x}, \phi_c, z_{CD}) P(z_{CD} | \mathbf{x}, \phi_c), \tag{4.7}$$

where $P(z_{CD}|\mathbf{x}, \phi_c)$ is calculated as in 4.4. Formally, ϕ_c corresponds to the single-frame trajectory at which a contact is detected. However, in case of non-perfect proprio-perception or measurement capabilities, this may correspond to a trajectory trunk instead of a single frame.
The touch measurement z_{touch} is comprised of both contact point z_P and normal vector z_n. This is observed only when the robot actually touches the object, *i.e.* when $z_{CD} = C$:

$$P(z_{touch} | \mathbf{x}, \phi_c, z_{CD}) = P(z_P, z_n | \mathbf{x}, \phi_c, z_{CD} = C). \tag{4.8}$$

In case of non-spherical end-effector, in [16], techniques have been presented to estimate the geometric parameters of the contacted surface through continuous force-controlled compliant motions. Since our final application case is comprised

of harsh scenarios presenting rough and possibly soiled surfaces, we may not assume the robot to be capable of performing compliant motions in such environments. Instead, actions with the spherical pin used as a depth-probing finger are here considered as motion primitives. Unfortunately, with such actions, it is possible to experience contact states leading to inaccurate measurements, *e.g.* when the pin touches the object over a vertex, an edge or a corner. In this case, the normal vector measurement is significantly biased with respect to the nominal vector of the adjacent faces, as depicted in Figure 4.4.

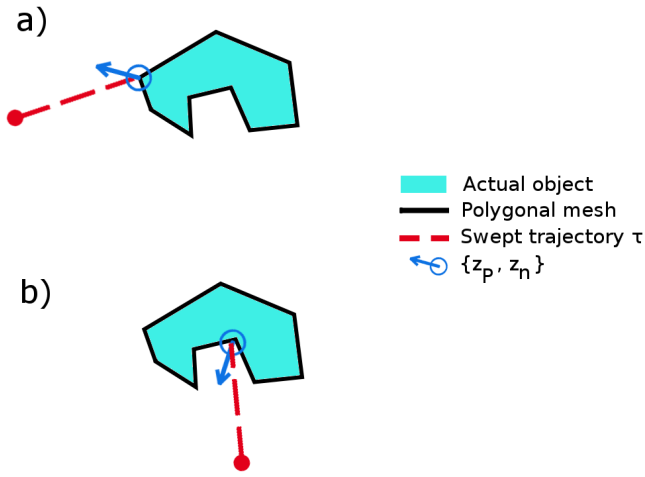

Figure 4.4: Examples of contact state possibly leading to a not accurate measurement: the end effector touches a vertex (case a) or a corner (case b). The measured normal does not correspond to any of the adjacent faces of on the object.

As in [28, 56, 32], we adopt the simplifying approximation of supposing both measurements z_P and z_n independent if conditioned on $\mathbf{x}, \phi_c$ and z_{CD}. So, the joint measurement probability can be expressed as in Eq. 4.9.

$$P(z_P, z_n|\mathbf{x}, \phi_c, z_{CD}) = P(z_P|\mathbf{x}, \phi_c, z_{CD})P(z_n|\mathbf{x}, \phi_c, z_{CD}). \tag{4.9}$$

With the object modelled as a polygonal mesh $\{f_i, n_i\}$ in pose $\mathbf{x}$ and with the robot in pose ϕ_c, one can express the likelihood functions encoding the probability of each patch $< f_i, n_i >$ to cause z_P and z_n as in Eq. 4.10 and Eq. 4.11. Specifically, the uncertainty on z_{touch} is due to the noise acting on the six channels of the force-torque signal, to vertex or edge contact interaction (as

depicted in Figure 4.4) and surface imperfections. Such noise is assumed to be Gaussian for both z_P and z_n.

$$P(z_P|f_i, \phi_c, C) = \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left[-\frac{1}{2} \frac{dist(z_P, f_i)^2}{\sigma_p^2} \right]. \tag{4.10}$$

and

$$P(z_n|n_i, \phi_c, C) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2} \frac{||z_n - n_i||^2}{\sigma_n^2} \right]. \tag{4.11}$$

The operator $dist(z_P, f_i)$ returns to the distance between a measured point z_P and a face f_i of the polygonal mesh, whereas $||z_n - n_i||$ is the norm of the difference between the measured normal vector and the patch normal vector (see Fig. 4.5).

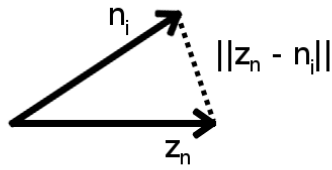

Figure 4.5: Norm of the difference between the nominal n_i and measured z_n normal vector. The error is modelled as affected by Gaussian noise, see Eq. 4.11.

The likelihood functions in Eq. 4.10 and 4.11 are written with respect to a single $< f_i, n_i >$ patch. Since it is necessary to express the likelihood functions with respect to a pose $\mathbf{x}$, there remains the problem of associating face-normal measurement probabilities to a pose. A maximum-likelihood approach can then be applied: the patch $< f^*, n^* >$ that maximises the product of both the contact and normal likelihoods is the one used to calculate the likelihood, as in Eq. 4.12.

$$< f^*, n^* >= \arg \max_{<f_i, n_i>} (P(z_P|f_i, \phi_c, C)P(z_n|n_i, \phi_c, C)). \tag{4.12}$$

This association strategy is also referred to as *hard assign* [36], and it implies an over-estimation of the likelihood function for the pose $\mathbf{x}$:

$$P(z_{touch}|\mathbf{x}, \phi_c) = (P(z_P|f^*, \phi_c, C)P(z_n|n^*, \phi_c, C)). \tag{4.13}$$

Alternatively, one may also follow a *soft assign* approach, considering the likelihood of the pose as a linear combination of the likelihood of its faces. Since computing Eq. 4.10 and 4.11 proves to be computationally expensive, the latter option is neither adopted in literature nor in this work.
Nevertheless, even using Eq. 4.12, computing contact and normal probabilities may become cumbersome when using large meshes. Instead, as already proposed by [32], one may select the face that only maximises $P(z_P|f_i, \phi_c, C)$, as in Eq. 4.14.

$$f^* = \arg\max_{f_i}(P(z_P|f_i, \phi_c, C)). \tag{4.14}$$

This allows the estimator to speed up the calculation since Eq. 4.11 is computed only once, but it prevents the model from considering the normal likelihood when selecting the face. Typically, the contact measurement is affected by a smaller noise than the normal measurement, so considering only the contact to assign the face causes a smaller error than doing the opposite. However, this simplification may be considered legitimate when $\sigma_p \ll \sigma_n$ (for instance with a difference of orders of magnitude), but it is not acceptable if σ_n approaches σ_p. For the sake of clarity Figure 4.6 illustrates an example of object mesh with a measured contact and normal vector affected by Gaussian noise. Being the closest to the contact point, face f_2 is selected to represent the pose of the object in terms of likelihood function(Eq. 4.15), even if n_2 differs of about 45deg from the measured normal vector.

$$P(z_{touch}|\mathbf{x}, \phi_c, C) = P(z_P|f_2, \phi_c, C)P(z_n|n_2, \phi_c, C). \tag{4.15}$$

Hence, one should be aware that the correspondence policy in Eq. 4.14 may significantly affect the likelihood calculation.

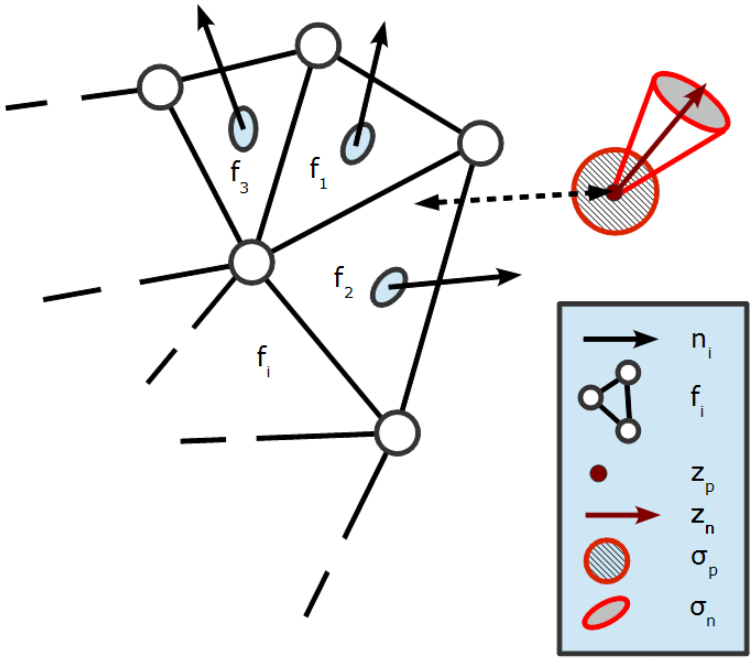

Figure 4.6: Polygonal mesh representing the object as a set of face-normal tuples $< f_i, n_i >$. Contact and normal vector are modelled as affected by Gaussian noise, represented by a sphere of radius σ_p and a cone of radius σ_n, respectively. Since the likelihood functions are formulated with respect to a single face (Eq. 4.10 and 4.11), the correspondance problem is solved as follows: the face that is closest to the contact point is the one used to calculate the likelihood for the whole object (Eq. 4.14).

4.2 DOF Decoupling Scalability With Rectangle Localisation Simulation

A first application of DOF Decoupling to a 3DOF active-sensing localisation is here presented. Inspired by the human experiment described in Chapter 2, we focus on steps 5 and 6 of the task sequence in Table 2.1, simulating the

localisation of a solid object positioned on top of a table. The demonstration has been implemented in Matlab. As in the experiment, the object is supposed fixed to the environment. In case of significant initial uncertainty (for instance [1m 1m πrad]), pose estimation may become cumbersome, suffering from the complexity of the configuration space. In addition, looking ahead to decide where-to-sense next requires a significant additional computational effort since simulating future measurements requires the update of the belief to compare the expected reward of each considered action. In the simplest case, when time horizon equals one, this corresponds to updating the posterior once for each element in the action space.
Since the main objective of this example is to show the effectiveness of DOF Decoupling in reducing the complexity of a high-dimensional problem, for the sake of simplicity, we choose to localise a rectangle whose position is bounded by the table sides. The rectangular shape allows us to analytically represent the geometry and compute the likelihood functions in Eq. 4.10 and Eq. 4.11. Nevertheless, the chosen decoupling scheme, measurement model and inference algorithm are rather generic, and can be applied to any object modelled with a polygonal mesh, regardless of its complexity. In this respect, examples are presented in the following sections with everyday industrial objects.

4.2.1 Task Description and Problem Complexity

The task is to localise a rectangle Q which can assume any $\{x, y\}$ position on top of a table, and can have any orientation θ_z around the z axis. The pose $\mathbf{x} = \{x, y, \theta_z\}$ is expressed with respect to the robot reference frame. Figure 4.7 illustrates the experiment set up with the table positioned in front of the robot R.
We suppose the robot to be equipped with a spherical end effector coupled with a force-torque sensor. R has to localise Q exploring the environment only with its end effector without contacting the environment with the rest of its structure. Information is acquired both while sweeping without contact and when the end effector establishes a contact with the object. We hypothesise the robot equipped with an ideal contact detector z_{CD} triggered any time the volume of the end-effector touches the object. When this happens, contact point and normal vector measurements $\{z_P, z_n\}$ are collected. The adopted measurement model is the one described in the previous Section (Eq. 4.4, Eq. 4.9, Eq. 4.10 and Eq. 4.11). Hence, the localisation task is to infer the pose of the object $\mathbf{x}$ given a measurement z. Applying the Bayes rule, we obtain:

$$P(\mathbf{x}|z) = \eta P(z|\mathbf{x})P(\mathbf{x}) \tag{4.16}$$

where η is a normalisation factor such that:

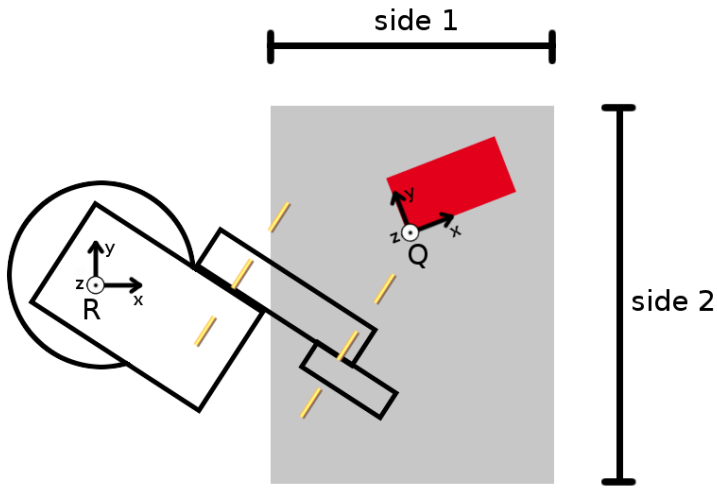

Figure 4.7: Set up for the rectangle localisation on top of the table

$$\int_X P(\mathbf{x}|z)dX = 1. \tag{4.17}$$

Bayesian parametric filters such as the Kalman Filter are not suitable for touch-based global localisation tasks such as the one presented in this work, since measurement equations are not linear and the robot does not know *a priori* which face of the object it is touching. Instead, sampling-based models such as the Histogram Filter or the Particle Filter allow us to treat the multi-modal nature of the posterior distribution in Eq. 4.16 discretising the configuration space, without making assumptions on the linearity of the measurement models [21, 28, 56, 32]. However, they require the calculation of the likelihood functions and the posterior update for each sample (particle), so the problem complexity is exponential in the number of degrees of freedom and its scalability in case of high initial uncertainty becomes a major issue. For example, adopting a Histogram Filter to discretise the posterior distribution on the configuration space $\{x, y, \theta_z\}$,

the number of nodes N is expressed in Eq. 4.18, with $\{res_x, res_y, res_{\theta_z}\}$ being the grid resolutions on the x, y, θ_z axes.

$$N = \frac{\text{range}(x)}{res_x} \times \frac{\text{range}(y)}{res_y} \times \frac{\text{range}(\theta_z)}{res_{\theta_z}}. \tag{4.18}$$

In order to illustrate the complexity of the state-estimation process, Figure 4.8 presents the number of grid nodes with range(x) = range(y) = 1m in translation and a resolution of $res_x = res_y = 5$mm. The exponential curves correspond to the number of nodes required with different values for res_{θ_z}, with range$(\theta_z) = 180$deg. With the update step requiring a computational time in the order of $\approx 10^{-4}$s for a single node, soon the overall update operation becomes computationally cumbersome.
On the other hand, untrained human beings have shown impressive skills in performing a 3DOF localisation task. For instance, the test presented in Chapter 2 took between 10s to 90s. During this time, the experimenters made decisions about where-to-sense-next, executed a number of actions and processed the collected information to accomplish the task. Nowadays, these numbers represent a challenging benchmark to match, with human beings outperforming robots in terms of action, perception and estimation time efficiency.

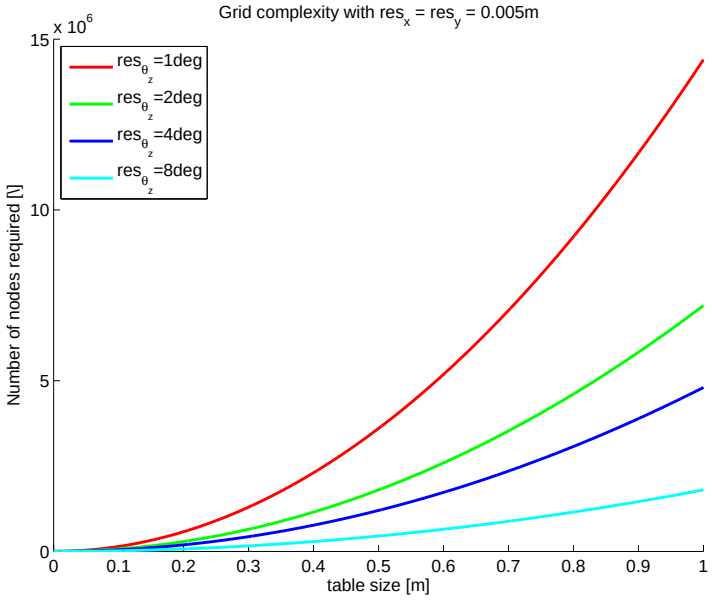

Figure 4.8: Complexity of 3DOF configuration space with x and y resolution fixed to 0.005m. The exponential curves correspond to different resolutions for the discretisation of θ_z.

4.2.2 Task Graph

Further to the human experiment presented in Chapter 2, the applied task graph is inspired by the common pattern among the subjects of moving from fast and imprecise sweeps before contact to fine and accurate motions after contact. Our intuition is to reduce the problem complexity by dividing the task into a first 2DOF subtask followed by a 3DOF subtask, in the framework of the DOF Decoupling Methodology. For similar applications, the scaling series algorithm [56] proves effective in inferring the pose of a solid object online ($\approx$1s), with initial uncertainty in the order of 0.5m in translation and πrad in rotation. The algorithm processes contact and normal-vector measurements through a series of successive refinements coupled with annealing. However, this method assumes the contact information to be available before running the estimation. Instead, our objective is to define a technique that allows to make decision *while* performing the localisation, so partial information can be immediately plugged in to decide where-to-sense-next. In addition, in our application, we are to consider both positive, *i.e.* contact and normal vector, and negative information, *i.e.* the volume swept by the robot without contact, tackling problems of initial uncertainty in the order of 1m x 1m x πrad.

Specifically, we select a DOF Decoupling task graph in which the task is divided into two subtasks: S1 and S2, defined over two configuration spaces C1 and C2. The applied scheme is taken as-is from the one empirically observed during the experiment presented in Chapter 2. Figure 4.9 illustrates the applied decoupling scheme under the graphical conventions of Figure 3.5. Subtasks S1 and S2 are described below under the DOF Decoupling formalism.

- **Subtask S1**:
  - *configuration space*: $\{x, y\}$ position of the object with respect to the robot
  - *information space*: posterior distribution over the object configuration space given the contact measurement
  - *action space*: sweep with contact detection
  - *inference scheme*: Histogram Filter
  - *action-selection scheme*: zig-zag motions parametrised with respect to the table dimensions.
- **Subtask S2**:
  - *configuration space*: $\{x, y, \theta_z\}$ position of the object with respect to the robot
  - *information space*: posterior distribution over the object configuration space given the contact measurement
  - *action space*: sweep with contact detection, contact-point and normal-vector estimation
  - *inference scheme*: Particle Filter
  - *action-selection scheme*: random actions aiming at the four faces of the object in its best-estimate configuration.

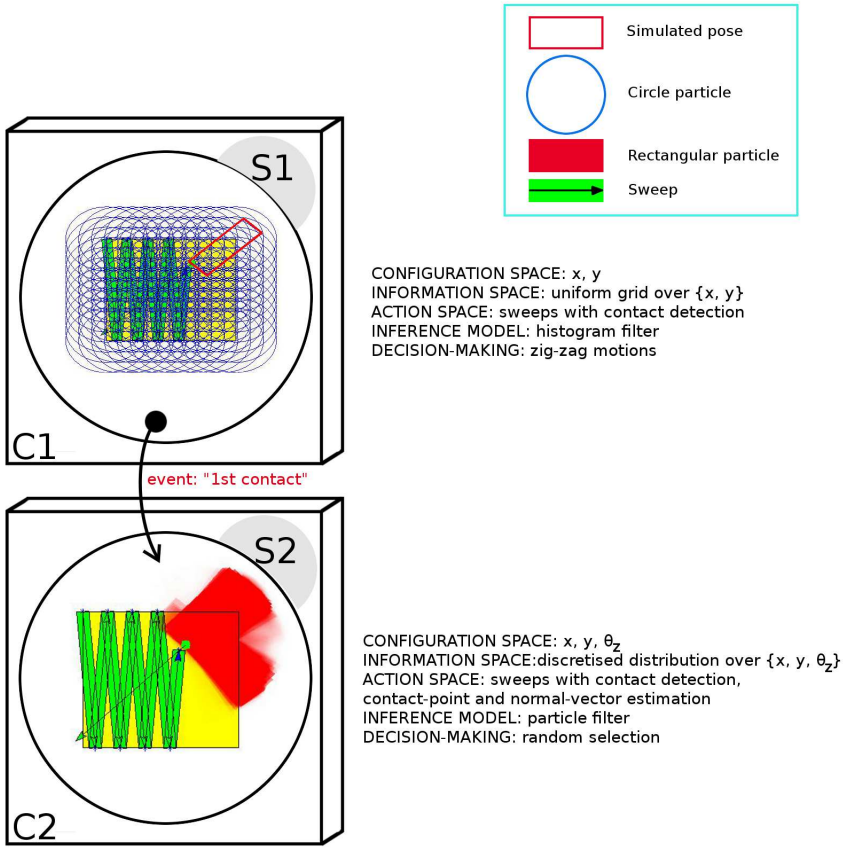

Figure 4.9: Task graph used for the rectangle-localisation case study. Inspired by the human experiment in Chapter 2, it is comprised of an initial 2DOF subtask (S1) defined over configuration space C1 and a second subtask (S2) defined over configuration space C2.

S1 is aimed at reducing the uncertainty on x and y applying a zig-zag action-selection strategy that spans the table surface in a breadth-first fashion. In order not to miss the object during each sweep, depth increments corresponds to the shortest dimension of the rectangle. The configuration space is discretised using a uniform grid over $\{x, y\}$ and the object is represented as the bounding circle of the rectangle with radius ρ, as depicted in Figure 4.10. In practice, each circle represents a node of the $\{x, y\}$ grid and all the possible θ_z rotation of the rectangle on that node. In S1, the action space is comprised of sweeps

with contact detection. The measurement model reflects the abstraction on the degrees of freedom, as the false-negative coefficient α in Eq. 4.4 is a function of the penetration distance $d(\mathbf{x}, \tau)$, which represents the smallest distance between the centre of the circle and the swept trajectory. This is used to capture the false-negative behaviour depicted in Figure 4.3. Among other geometry-penetration metrics, it is convenient to consider a point-to-segment distance for computational reasons.

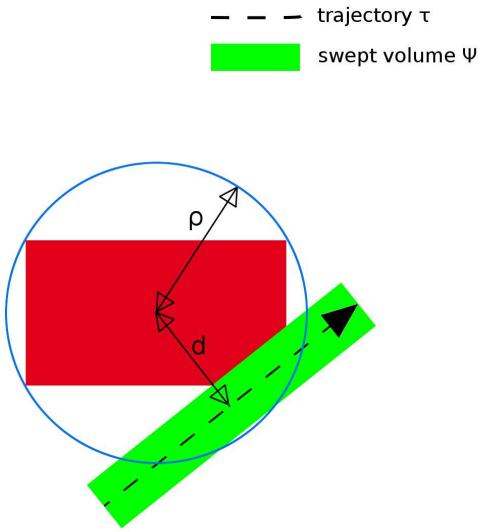

Figure 4.10: Geometric representation of the object in S1 (blue bounding circle of radius ρ) and in S2 (nominal rectangle). $d(\mathbf{x}, \tau)$ denotes the distance between the centre of the circle and the sweep trajectory.

Intuitively, given a sweep without contact, the deeper it penetrates a circle particle, the less the configurations represented by such particle are likely to correspond the actual configuration of the object. Specifically, we take into account all the configurations the rectangle can assume inside the circle, that is $\theta_z \in [0, \pi]$, and we uniformly discretise the penetration depth d which can be swept by the end effector:

$$d \in \left[0, \rho + \frac{\kappa}{2}\right]. \tag{4.19}$$

Hence, α corresponds to one minus the ratio between the rectangles intersected by the sweep n_{int} and the total number of rectangles n_{tot} used to discretise

θ_z. Eq. 4.20 expresses the false-negative coefficient α as a function of the penetration distance d:

$$\alpha(d) = 1 - \frac{n_{int}(d)}{n_{tot}}. \tag{4.20}$$

It is important to note that different dimensions of the rectangle and sweep thickness κ result in different $\alpha(d)$ functions. To speed up the run-time calculation, a lookup table can be pre-computed offline for Eq. 4.20. Figure 4.11 presents an example of intersected rectangles at three different values of d. Figure 4.12 and Figure 4.13 illustrate the $P(NC|\xi(\mathbf{x}, \tau) = 1)$ and $P(C|\xi(\mathbf{x}, \tau) = 1)$ models for a rectangle of dimensions 0.1m x 0.2m and $\kappa = 0.05$m. Both functions are computed up to the intersection limit, that is the maximum value of d implying a superposition between the sweep area and the circle.

According to the task graph depicted in Figure 4.9, the first contact triggers the transition to S2, which focuses on refining the estimation on x, y, adding the extra degree of freedom θ_z to the configuration space. In S2, the object is represented by its nominal rectangular geometry, the action space is comprised of sweeps with contact detection and touch measurement, a Particle Filter is used for the inference and actions are randomly selected aiming at the four faces of the best estimate. In order to transfer the knowledge from S1 to S2, samples on the x and y coordinates are drawn from the 2-dimension distribution inherited from S1. Specifically, samples are drawn from a uniform distribution on the rotation θ_z, and from a 2D multivariate Gaussian Mixture with kernels centred on the 2D grid and diagonal covariance matrix Σ. To take into account the location of the circle particles, the eigenvalues of Σ are set equal to the squared inter-nodal distance of the grid. For further details about sampling techniques we refer to [26]. The presented example was implemented using Matlab's Statistical Toolbox.

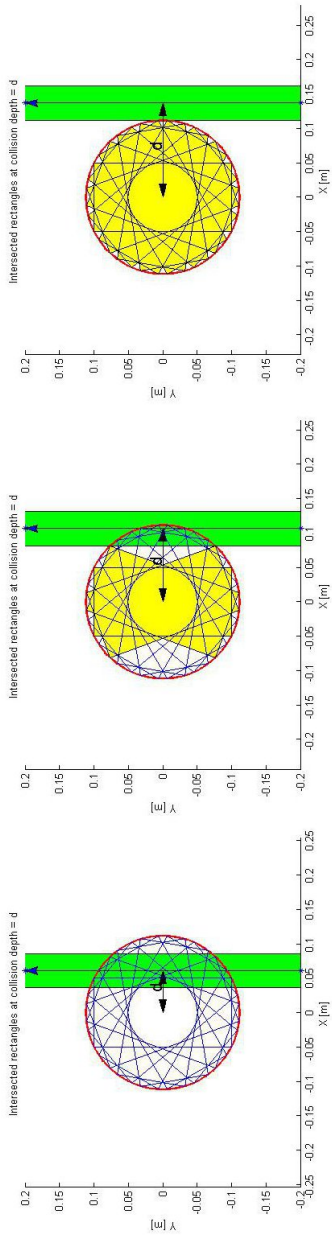

Figure 4.11: Intersected rectangles with different collision depth used to compute the $\alpha(d)$ lookup table. The end-effector sweep is depicted in green, the rectangles which are not crossed are coloured in yellow. For each value of the penetration distance d, $\alpha(d)$ is set equal to the number of not intersected rectangles. In this figure, for the sake of clarity, only ten rectangles are depicted.

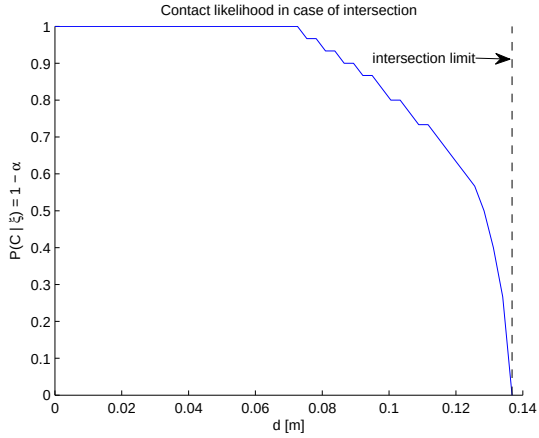

Figure 4.12: Contact likelihood in case of intersection as function of the collision depth. $\alpha(d)$ depends on the sweep thickness κ and the dimensions of the rectangle. Hence, a lookup table is built offline computing $\alpha(d)$ discretising d uniformly over $[0, \rho + \frac{\kappa}{2}]$.

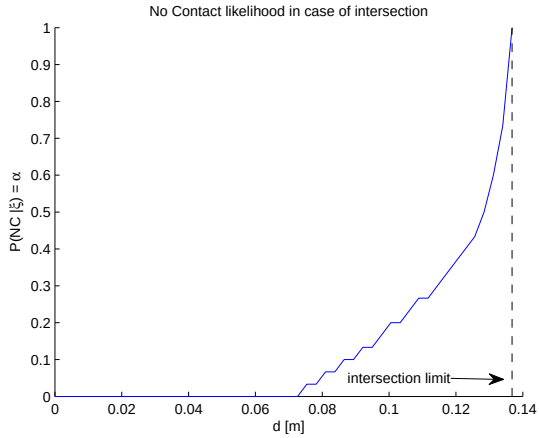

Figure 4.13: No-contact likelihood in case of intersection as function of the collision depth. $\alpha(d)$ depends on the sweep thickness κ and the dimensions of the rectangle. Hence, a lookup table is built offline computing $\alpha(d)$ discretising d uniformly over $[0, \rho + \frac{\kappa}{2}]$.

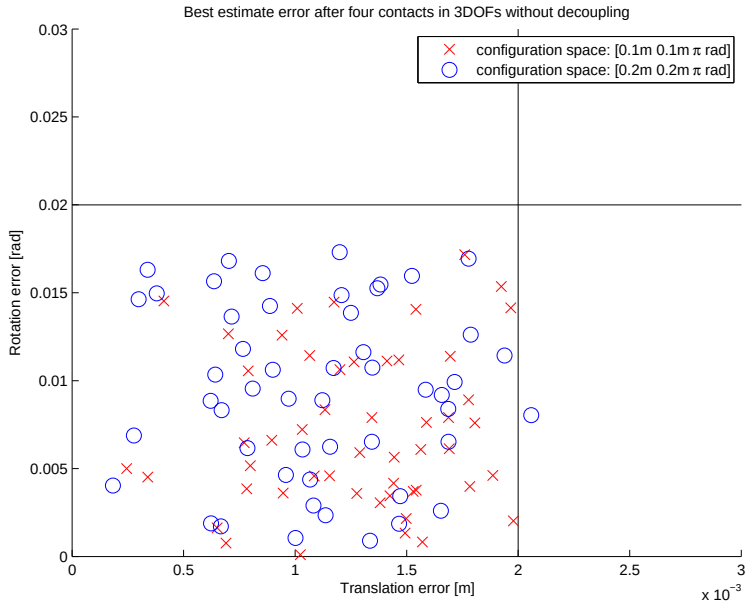

Figure 4.14: Final error of best estimate when 90% of probability mass within 5mm. Tests are carried out with ideal measurements and two initial uncertainties: [0.1m, 0.1m πrad] and [0.2m, 0.2m πrad]. The resolutions $[res_x\ res_y\ res_{\theta_z}]$ are illustrated with black lines.

4.2.3 Measurement Model Validation

The measurement model presented in Section 4.1 is validated by localising the rectangle in 3DOFs ($\{x, y, \theta_z\}$) without decoupling, with low initial uncertainty in order to keep the problem tractable computationally. Contact point and normal vector noises used to calculate Eq. 4.10 and Eq. 4.11 are set to $\sigma_P = 0.005$m and $\sigma_n = 0.1$ and the measurements are collected on the four faces of the object. Twenty runs with the object located in a random $\{x, y, \theta_z\}$ pose were simulated with initial uncertainty set to [0.1m, 0.1m, πrad] and [0.2m, 0.2m, πrad]. Initially, the end-effector sweeps the configuration space with a zig-zag trajectory parametrised with respect to the dimension of the table. After the first contact is established, actions aimed at touching the four faces of the current best estimate are executed in random order. The task is considered accomplished when 90% of the probability mass of the target position is concentrated in a circle of radius = 5mm. In this case, the target corresponds to the center of the rectangle.
Figure 4.14 illustrates the final translation and rotation error when the

probability mass exceeded the 5mm threshold. This corresponds to the misalignment between the best estimate and the simulated object due to the limited resolution ($\{res_x, res_y, res_{\theta_z}\}$) of the grid. In both cases the estimation results in the best estimated error within the grid tolerance, so the measurement model and inference scheme work as expected.

4.2.4 DOF Decoupling Benefits

The task graph in Subsection 4.2.2 is applied to the rectangle localisation task. The objective of this test is to evaluate its performance in terms of reliability and computational efficiency, with respect to those achieved without decoupling. First, the decoupling scheme S1→S2 is applied to a localisation task with initial uncertainty of [0.8m 0.8m πrad], which corresponds to a configuration space 16 times bigger than the one treated in the previous section without decoupling. In S1, the grid is generated with $res_x = res_y = 0.005$m. Simulated measurements are biased with an artificial noise sampled from Gaussian distributions with zero-mean and standard deviations $\sigma_P = 0.005$m and $\sigma_n = 0.1$. Figure 4.15 presents the final estimation error obtained over 20 runs with 40000 particles used in S2. Actions were performed until 90% of the probability mass was concentrated within a 5mm radius. Comparing Figure 4.15 to Figure 4.14, we observe an increase of both the translation and rotation final error: this is not surprising as the Figure 4.15 reports localisation results biased by simulated noise and the loss of information possibly occurring in the transition from S1 to S2.

In order to highlight the benefit of DOF Decoupling, Figure 4.16 presents a comparison of the computational time required to process both the contact and no-contact measurements using the 3DOF grid method and the S1→S2 task-graph with a square table of increasing dimensions. As expected, the computational time recorded withoud decoupling scales approximately with the cube of the table side. Run time with DOF Decoupling is instead proportional to the square of the table side. It is important to mention that these computational times are obtained with a non-optimised Matlab implementation. They are here reported to describe the scale of the problem with and without decoupling. Nevertheless, a significant performance improvement is expected with an optimised C++ implementation.

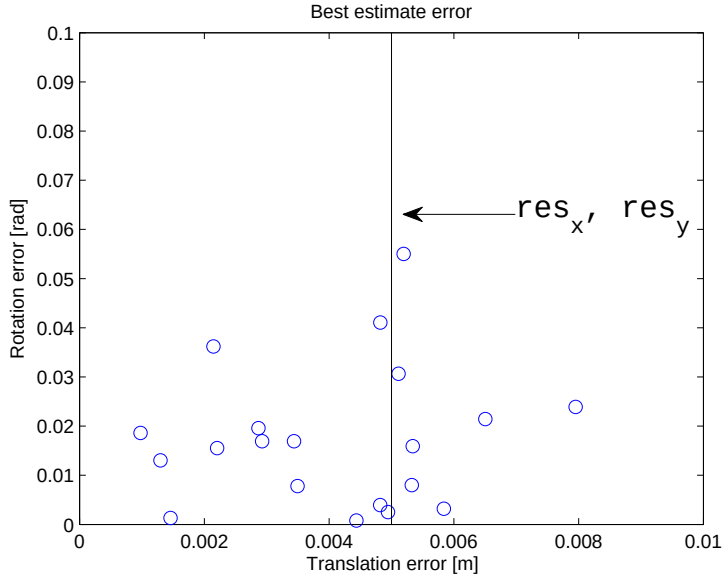

Figure 4.15: Final error with DOF Decoupling with simulated measurement noise $\sigma_P = 0.005$ and $\sigma_n = 0.1$. Initial uncertainty = [0.8m 0.8m πrad]. The vertical black line represents the x and y tolerances of the uniform grid used in subtask S1.

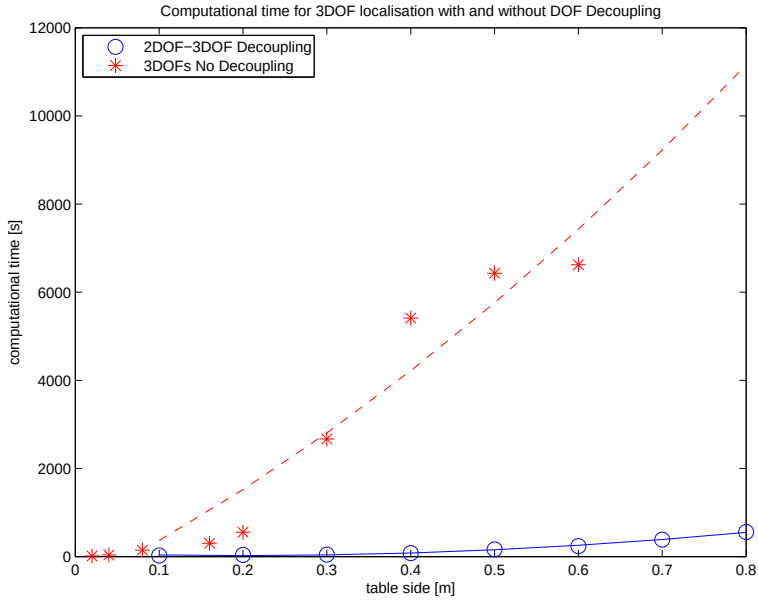

Figure 4.16: Computational time comparison for 3DOF localisation with and without DOF Decoupling.

4.3 Robot Experiment: Localisation of Industrial Objects of Increasing Complexity

The measurement model presented in Section 4.1 has been applied to localise a set of industrial objects of increasing complexity with a Staubli RX90 robot. The aim of these tests is to show the general applicability of such methodologies, enabling the robot to localise everyday objects, regardless of their size and shape, up to items of industrial relevance. To the best of our knowledge, literature examples presented case studies with polygonal meshes of about 100 faces [56, 28]. Figure 4.17 presents a picture and the polygonal mesh of the objects used in the experiment: the solid rectangle (obj_{SR}), the v-block (obj_{VB}), and the ergonomy test mock-up (obj_{ET}). Table 4.1 presents their geometrical and dynamical properties. The robot is equipped with a spherical end-effector that is coupled with a force-torque sensor. Made of aluminum, its shape was designed finding a compromise between stiffness and weight. It is represented in Figure 4.18 and its blueprint is in Figure 4.19.

Object	Dimensions[m]	Weight[kg]	Mesh Vertices	Mesh Faces
obj_{SR}	0.15 x 0.15 x 0.1	1.2	8	12
obj_{VB}	0.175 x 0.15 x 0.05	3	96	188
obj_{ET}	0.44 x 0.14 x 0.14	10.5	915	1877

Table 4.1: Geometric and dynamic properties of the three objects used in the experiment also depicted in Figure 4.17.

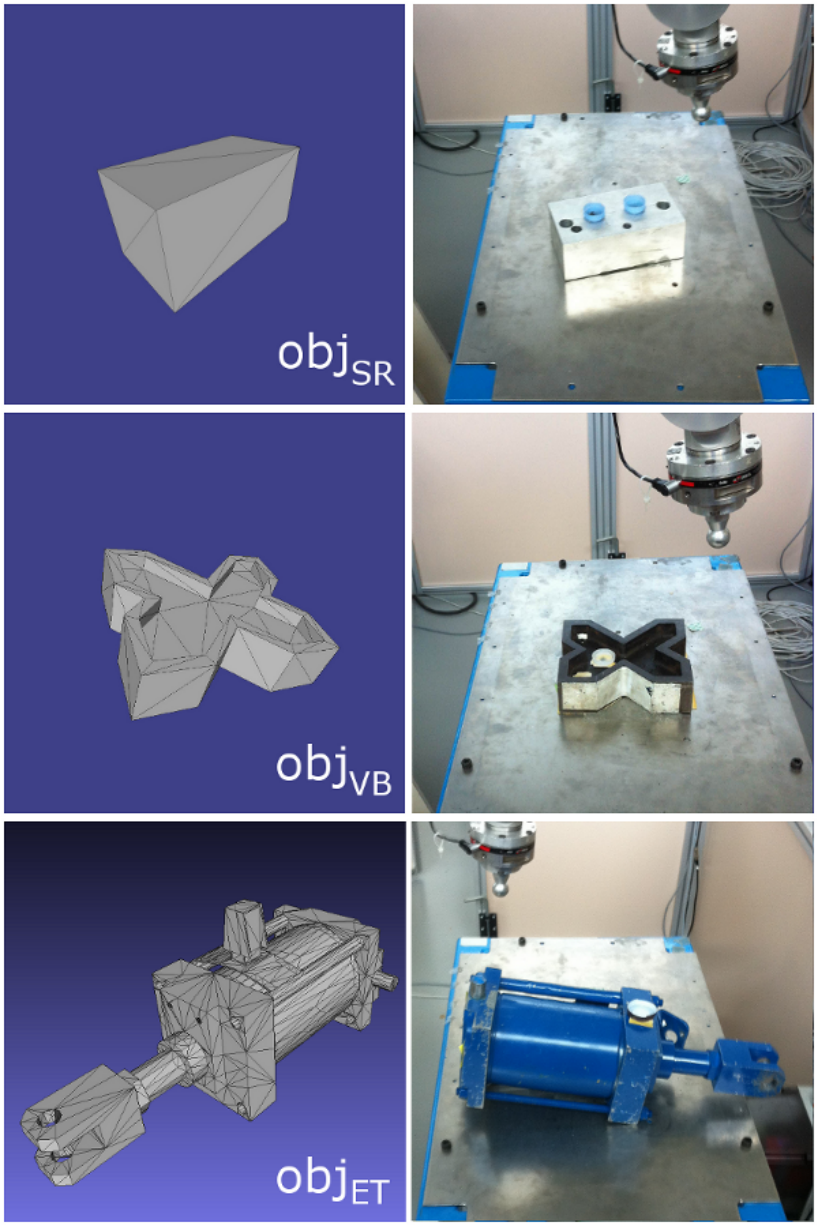

Figure 4.17: The three objects used as case-study examples : (1) solid rectangle obj_{SR} modelled with 8 vertices, (2) v-block obj_{VB} modelled with 96 vertices and (3) the ergonomy test mock-up obj_{SR} modelled with 915 vertices.

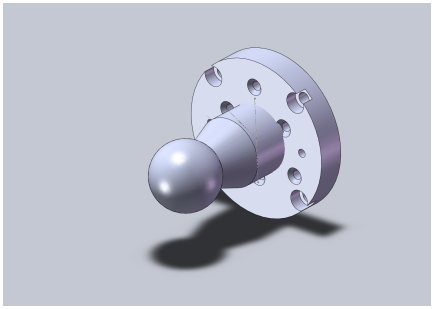

Figure 4.18: The spherical end-effector used in the experiments.

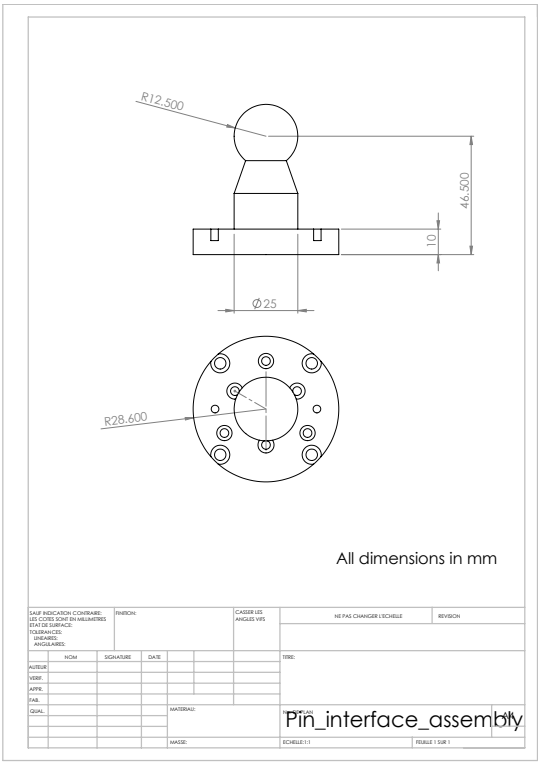

Figure 4.19: Blueprint of the spherical end-effector used in the experiments.

4.3.1 Measurement System

In order to identify the measurement standard deviation errors σ_p and σ_n in Eq. 4.10 and in Eq. 4.11, a sequence of repeated contact tests has been carried out setting the force threshold at 5N, 10N, 20N, 30N, 40N and 50N. The robot was controlled in PID mode with a setpoint located beyond one of the faces of the obj_{VB}. For each value, 30 contacts have been repeated on the same position in order to monitor the repeatability of the measurement. Results are presented in Figure 4.21 and Figure 4.22.
The collected measurements show a repeatability with standard deviation below 0.002m even for the 5N case, which is consistent to the level of resolution required for object manipulation. However, even though the object was glued to the table, a significant offset was recorded between trials with different thresholds. After measuring negligible relative displacement between the object and the table, it is reasonable to assume that such behaviour is due to a displacement of the table during contact. Hence, the inference model has to take into account the additional uncertainty caused by such displacement. In a Bayesian context, this is captured by the transition model $P(\mathbf{x}_{t+1}|\mathbf{x}_t)$. In the experiment presented in this chapter, the uncertainty introduced by the contact is modelled as a multivariate Gaussian with diagonalised covariance matrix. Specifically, we set the translation and orientation covariance to $\sigma_x^2 = \sigma_y^2 = 10^{-6}\text{m}^2$ and $\sigma_{\theta_z}^2 = 2.5\,10^{-3}\text{rad}^2$.

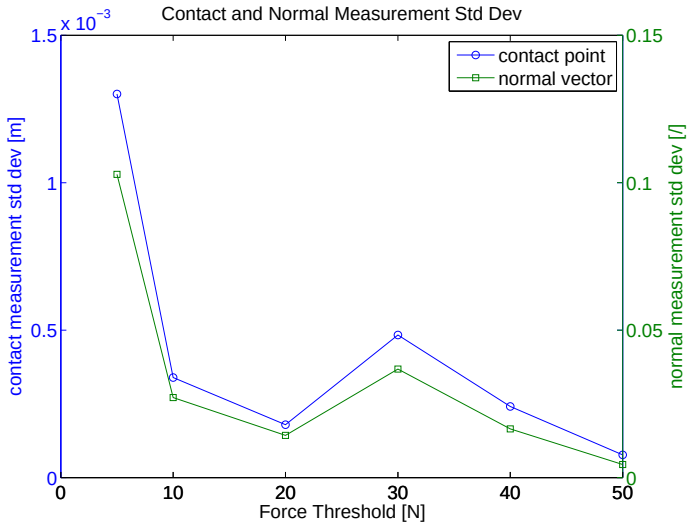

Figure 4.20: Standard deviation of the contact-point and normal vector measurement as function of the force threshold to detect contact.

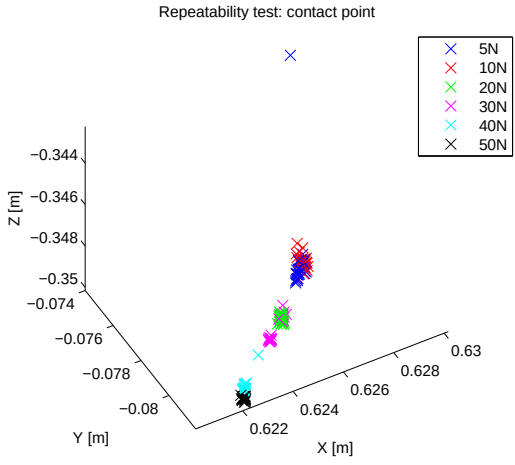

Figure 4.21: Repeatability test: contact point estimation with different values of the contact detection threshold. An outlier measurement was recorded during the 5N trial.

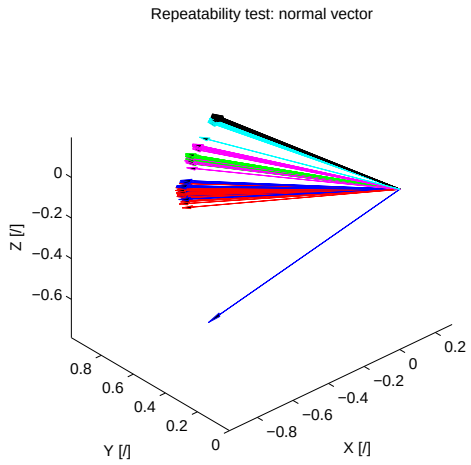

Figure 4.22: Repeatability test: normal vector estimation with different values of the contact detection threshold. An outlier measurement was recorded during the 5N trial.

4.3.2 Inference Scheme

The state $\mathbf{x}_t \in \{x, y, \theta_z\}$ represents the pose of the object with respect to the robot frame of reference at time t. As observed in the contact trials presented in Subsection 4.3.1, the displacement occurring during probing does not allow to assume the object to be fixed, so the evolution of its pose over time is described as a Hidden Markov Model, *i.e.* a dynamical stochastic system where $\mathbf{x}_t$ depends only on the previous state $\mathbf{x}_{t-1}$. This derives from the Markov assumption of state completeness, *i.e.* no variable prior to $\mathbf{x}_{t-1}$ may influence the evolution of the future states. The state-measurement model can then be summarised as:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}) + \delta_t \tag{4.21}$$

$$z_t = m(\mathbf{x}_t) + \epsilon_t \tag{4.22}$$

where $g(.)$ and $m(.)$ are the transition model and measurement functions and δ_t and ϵ_t are transition and measurement noises. In a Bayesian context, the state is inferred updating a posterior distribution over the configuration space accounting for the measurement z_t which is modelled as a set of independent random variables z_t drawn from the conditional probability distribution $P(z_t|\mathbf{x}_t)$, also referred to as likelihood. The motion model is encoded by the transition probability $P(\mathbf{x}_t|\mathbf{x}_{t-1})$. Our estimation objective is to infer the posterior probability over the state given the available measurement $P(\mathbf{x}_t|z_{0:t})$. In this

work, the notation proposed in [63] is taken as reference to represent the state transition and update. Specifically, $\bar{bel}(\mathbf{x}_t)$ and $bel(\mathbf{x}_t)$ are used to denote these two conditional probabilities, as in Eq. 4.23 and Eq. 4.24.

$$\bar{bel}(x_t) = P(\mathbf{x}_t | z_{0:t-1}). \tag{4.23}$$

$$bel(x_t) = P(\mathbf{x}_t | z_{0:t}). \tag{4.24}$$

A Bayesian filter requires two steps to build the posterior probability, a prediction and an update. In the former, the belief is calculated as in Eq. 4.25, projecting the state from time $t-1$ to time t using the transition model.

$$\bar{bel}(x_t) = \int_{\mathbf{x}_{t-1}} P(\mathbf{x}_t | \mathbf{x}_{t-1}) \bar{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \tag{4.25}$$

In the latter, the posterior probability is updated multiplying the belief $\bar{bel}$ by the measurement likelihood and the normalisation factor η. Intuitively, this corresponds to a *correction* of the model-based prediction based on measured data:

$$bel(x_t) = \eta P(z_t | \mathbf{x}_t) \bar{bel}(\mathbf{x}_t). \tag{4.26}$$

In our application, the measurement likelihood $P(z_t|\mathbf{x}_t)$ is calculated as in Eq 4.5 and Eq. 4.6. The normalisation factor is calculated as

$$\eta = \frac{1}{\int_{\mathbf{x}_t} P(z_t | \mathbf{x}_t) \bar{bel}(\mathbf{x}_t) d\mathbf{x}_t} \tag{4.27}$$

and assures that the posterior probability sums up to one. Since we may not assume the posterior distribution over the state to be unimodal and the measurement is not linear (Eq. 4.10 and Eq. 4.11), a Particle Filter algorithm is adopted. Its implementation is presented in pseudo code in Algorithm 2, with χ representing the particle set. χ_0 is initialised with samples drawn uniformly over the configuration space. Every time an action is completed, the algorithm is run. In case of contact, the weights are calculated according to Eq. 4.6. In case of trajectory finished without contact, the weight is updated according to Eq. 4.5. Further details on the Particle Filter model can be found in the Appendix.

4.3.3 Action Space and Action-Selection Scheme

As in [28, 32], actions are comprised of trajectories $\{\tau\}$ defined relatively to the object geometry aiming at touching its lateral faces moving parallel to their normal vector. Since the sensing apparatus is the one described in Section 4.1,

Algorithm 2 Object Localisation With Particle Filter

$\chi_t = []$
for $i = 1 : N$ **do**
  sample $x_t^{[i]} \sim P(x_t | x_{t-1}^{[i]})$ ▷ Draw particle from transition model
  if contact **then** ▷ update contact weight
    $w_t^{[i]} = P_{free}(x_t^{[i]}, \tau_c) P(C, z_{touch} | x_t^{[i]}, \phi_c)$
  else ▷ Update no-contact weight
    $w_t^{[i]} = P_{free}(x_t^{[i]}, \tau)$
  end if
  add $x_t^{[i]}$ to $\bar{\chi}_t$
end for
for $i = 1 : N$ **do**
  draw index m from $\{x_t^{[i]}, w_t^{[i]}\}$ ▷ Resampling with replacement
  add $x_t^{[m]}$ to χ_t
end for
for $i = 1 : N$ **do**
  $\tilde{w}_t^{[i]} = \frac{w_t^{[i]}}{\sum_0^N w_t^{[i]}}$ ▷ Normalisation
end for

according to Def. 1.6, there is a one-to-one correspondence between action and trajectory. Therefore, these two terms will be used a synonyms in the rest of the Chapter.
In addition, a target point is defined to conclude the localisation with a peg-in-hole motion. Figure 4.23 and 4.24 illustrate the actions defined for obj_{SR} and obj_{VB} together with the target points. A thorough analysis of the problem of defining sensing actions is presented in Section 4.4.1, focusing on the decision-making side of active sensing. In the current experiment which focuses on how to treat information, the set of available actions $\{\tau_i\}$ is considered pre-defined and is specified by the robot programmer. At run time, actions are drawn randomly with respect to the best-estimate configuration.

4.3.4 Test Results

Tests have been carried out localising the objects $\{\text{obj}_{\text{SR}}, \text{obj}_{\text{VB}}, \text{obj}_{\text{ET}}\}$ with an initial uncertainty on $\mathbf{x}$ of [0.1m 0.1m 1rad]. A set of trajectories $\{\tau\}$ was defined relatively to the objects' geometry and a feature point was identified as target on each object. Trajectories and targets are illustrated in Figure 4.23, 4.24 and 4.25. The following protocol was followed for each run:

1. Actions were repeated until 80% of the probability mass on the target position was concentrated around a 5mm radius, which is consistent with

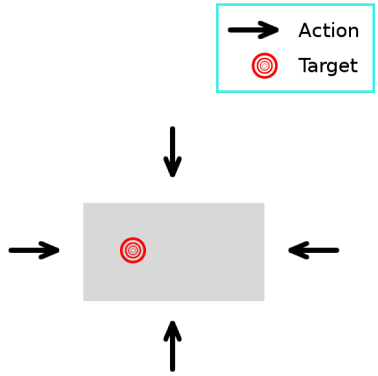

Figure 4.23: Actions defined relatively to the obj_{SR}

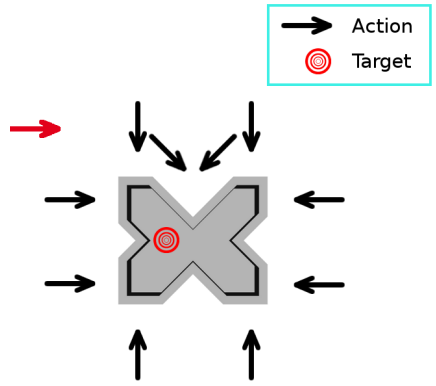

Figure 4.24: Actions defined relatively to the obj_{VB}.

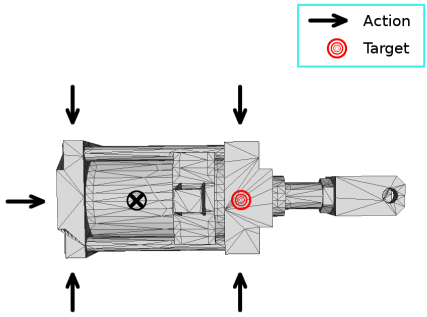

Figure 4.25: Actions defined relatively to the geometry of obj_{ET}.

the tolerance allowed in typical robot operations, *e.g.* for grasping and manipulation applications. The probability threshold was chosen to be the same as in [32].

2. Once the 80% confidence threshold was reached, a peg-in-hole sequence aiming at the target on the object was performed recording the final position of the spherical end effector.

3. The end-effector was manually positioned on the target spot and the end-effector position recorded.

The considered ground truth about the position of the target was biased by the uncertainty related to human vision. After empirical observation, this can be estimated on the order of 2mm. Even if biased, this may be considered as a genuine mean to evaluate the localisation accuracy. Figure 4.26, 4.27 and 4.28 present the localisation results in terms of final $\{x, y\}$ target position error with respect to the observed ground truth. The mean and the standard deviation of the error norm are also reported in the plots.
Overall, the tests carried out on the obj_{VB} present the best accuracy in terms of mean error norm, even though an outlier was recorded due to a vertex contact. This resulted in a significantly biased final estimate of the target, as shown in Figure 4.27. Unsurprisingly, tests on obj_{ET} present the highest mean and standard deviation error, yet with most of the bias concentrated along the x axis. This is likely to be due to a non-negligible misalignment between the polygonal mesh and the actual geometry of the object which experienced

plastic displacement due to heavy usage. In addition, during the experiment, we observed a more significant object displacement during the tests on obj_{SR} and obj_{ET} than those on obj_{VB}, due to a greater moment applied by the robot to the table during contact (the point of application of the contact force was higher with respect to the table surface).

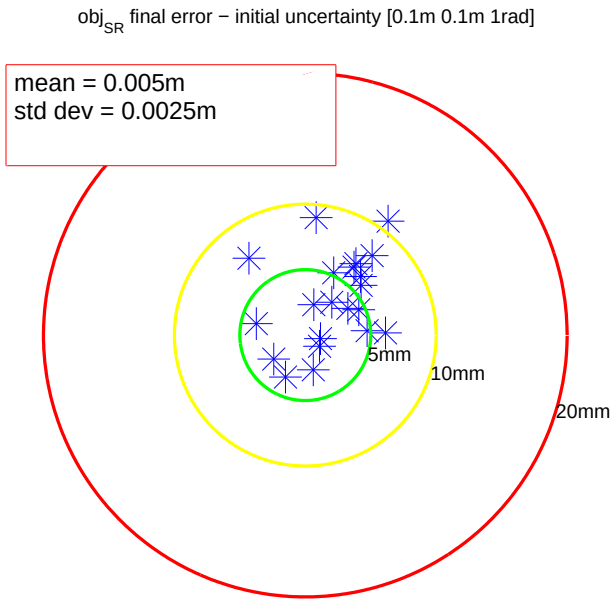

Figure 4.26: Localisation tests with obj_{SR}. Initial uncertainty [0.1m 0.1m 1rad]. The final error on the target position is measured against a human-observed ground truth obtained positioning the end effector on the target spot. Both the mean and standard deviation of the error norm are reported.

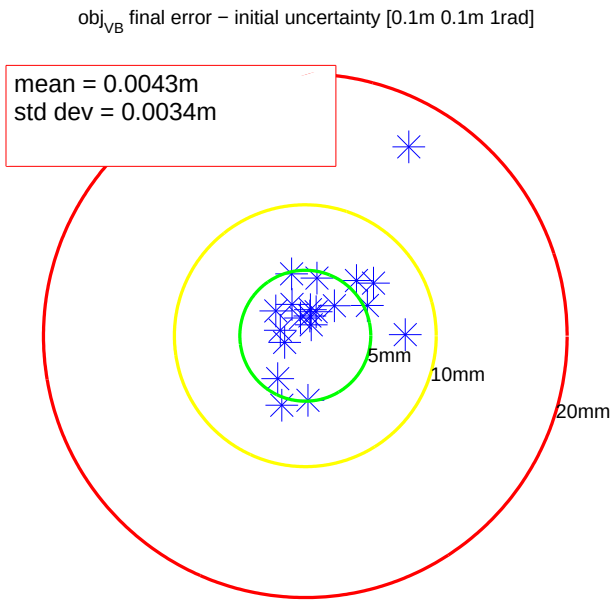

Figure 4.27: Localisation tests with obj_{VB}. Initial uncertainty [0.1m 0.1m 1rad]. The final error on the target position is measured against a human-observed ground truth obtained positioning the end effector on the target spot. Both the mean and standard deviation of the error norm are reported. An outlier estimate was recorded during the tests, as highlighted in the plot. This was due to a vertex contact which fed a misleading measurement to the estimator.

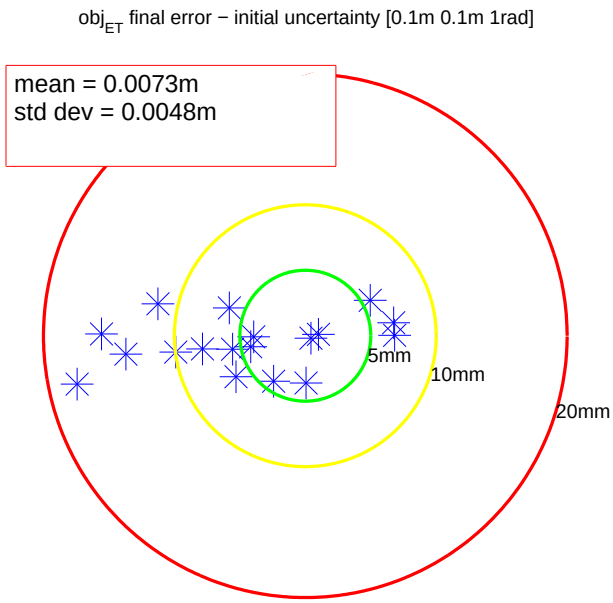

Figure 4.28: Localisation tests with obj_{ET}. Initial uncertainty [0.1m 0.1m 1rad]. The final error on the target position is measured against a human-observed ground truth obtained positioning the end effector on the target spot. Both the mean and standard deviation of the error norm are reported.

4.3.5 Computational Complexity

In the presented localisation application, the major computational effort is represented by the solution of the association problem in Eq. 4.14, and the distance calculations required by Eq. 4.10 and Eq. 4.11. The running time depends on the number of vertices of the polygonal mesh that represents the object. Specifically, the shortest-distance calculation to identify the maximum-likelihood face in Eq. 4.14 parses all the faces of the mesh to find the one that is the closest to the contact point. This operation is repeated for all the particles used for the inference, adopting the contact-detection suite included in the XDE® Physics Engine software package, developed by the Interactive Simulation Lab at CEA. Based on dilated simplicial complexes and bounding volumes hierarchies [47], queries between the contact point and meshes allows us to identify the closest point on the surface as in Eq. 4.14, thus identifying f_i and n_i to compute the likelihood functions in equation Eq. 4.10 and Eq. 4.11. To evaluate its performance with the three case-study meshes, trials have been done using 100, 1000 and 10000 particles. In order to speed up the calculation,

one can limit the query for the closest geometric entity up to a certain radius θ_z around the contact point. This results in a significant reduction of computational time, which can be quantified in a 10x factor for δ =0.005m. However, limiting the query corresponds to saturating all the distance beyond δ, possibly losing useful information. Table 4.2 summarises the results obtained for the three meshes with different number of particles reporting the time to compute the contact likelihood t_C and the time to compute the no-contact likelihood t_{NC}.

Vertices	Particles	t_C $(\theta_z = \infty)$[s]	t_C $(\theta_z = 0.005m)$[s]	t_{NC}[s]
8	100	0.02	0.01	0.01
8	1000	0.21	0.1	0.02
8	10000	1.35	1.25	1.03
96	100	0.08	0.02	0.01
96	1000	0.8	0.18	0.02
96	10000	8.2	1.74	0.2
915	100	0.58	0.02	0.02
915	1000	5.93	0.28	0.12
915	10000	60.06	4.88	1.18

Table 4.2: Computational time to calculate both contact t_C and no-contact t_{NC} likelihood with objects of increasing complexity. Reported running times are obtained with a C++ implementation running on a 2.16GHz double-core computer with 2Gb RAM.

4.4 Experimental Comparison of Action-Selection Schemes

Different action-selection schemes are here presented comparing their performance for localisation applications. Firstly, this Section illustrates how the elements composing the action space are generated by the robot. Then, possible metrics to quantify the information gain as reward function to select actions are presented in the context of touch-based active sensing. Finally, an experimental comparison of different action-selection schemes is carried out measuring their performance in terms of execution time to localise obj_{VB}.

4.4.1 Action Space Generation

From an active-sensing perspective, actions represent the primitives considered by the robot deciding where-to-sense-next. According to Def. 1.6, each action is comprised of a trajectory τ and the measurement z collected during the motion. This information is used to update the posterior distribution over the pose of the object. In the context of robot grasping, several works have

faced the problem of defining sensing actions in order to localise and/or grasp a known object. In [30], so-called world-relative trajectories were defined with respect to the geometric model of the object. The robot was equipped with a three-finger hand, so the space of the actions was huge in terms of possible trajectories. Thus, a small subset of actions was selected, including: a human-demonstrated goal action that grasps the object if executed with respect to the correct world state; information actions attempting to contact non-goal surfaces of the object; re-orientation actions that use a grasp from above to rotate the object and increase robot dexterity for the following actions. Since a specific object configuration had to be chosen to calculate the trajectories, at run time, they were transformed to adapt to the current best-estimate pose.
With a similar setup, in [28], the set of actions was defined as follows. First, surfaces with associated normal directions were identified on the polygonal mesh. Then, hand and finger trajectories were chosen aligning these directions with those of the palm and finger sensors. Additional motions could also be specified by the user. Actions in which the hand collides with other objects were pruned out of the candidate touching actions.
Building upon the aforementioned literature examples, in this work, actions are defined relatively to the object model represented by a polygonal mesh $\{f_i, n_i\}$. Polygonal faces larger than the maximum cross section of the used end-effector are identified as target faces $\{f_k, n_k\}$. Each action is then comprised of a reaching trajectory τ parallel to n_k aiming at the center of f_k. This strategy is inspired by the concept of affordances [11, 51] which links tool, motion and effect as an atomic primitive. This approach allows the robot to adapt the set of considered actions to the available setup. Intuitively, the same object would *afford* different $< face, trajectory >$ tuples if the sensory apparatus was different. During motion, the sensing apparatus operates as a contact detector device while during contact it estimates both contact point and normal vector, as described in Section 4.1. As in [30], actions are defined with respect to a single pose of the object. Formally, we will denote the action set as $A = \{\tau\}$. Any time the robot needs to decide where-to-sense-next, each action τ is transformed according to the best estimate of the pose of the object $\hat{\mathbf{x}}$.

4.4.2 Information Gain of an Action

Information gain (IG) is a measure of how much knowledge is brought in by a new action, given the belief on the state. In general, this is a function of the dispersion of the posterior probability on the state given the observation collected during the action. Different information-gain metrics will be presented in Subsection 4.4.3. In the context of action selection, it is important to note how a future measurement is only hypothetical, so we can only calculate the expected information-gain over the possible measurements. As in [31], here we

make the simplifying assumption of considering only touch measurement when computing the expected information gain of a future trajectory τ:

$$E[IG(\mathbf{x}, \tau)] \approx E[IG(\mathbf{x}, z_{touch})]. \tag{4.28}$$

We recall that a touch measurement is collected moving along τ, after a collision with the robot stopped in frame ϕ_c. Yet, such frame is not deterministically known, so we name $\tilde{\phi}_c$ a generic contact frame along τ. In addition, given $\tilde{\phi}_c$, touch measurement is hypothetical as well, and will be denoted as $\tilde{z}_{touch}$.
Hence, in order to estimate the information gain of an action, one has to calculate its expected value over the trajectory and the measurement space:

$$E[IG(\mathbf{x}, \tau)] = \int_{\tau} P(\tilde{\phi}_c = \phi | \mathbf{x}, \tau) \int_{Z} IG(\mathbf{x}, \tilde{z}_{touch}) P(\tilde{z}_{touch} | \mathbf{x}, \tilde{\phi}_c, C) dz d\phi. \tag{4.29}$$

The calculation of the integral in Eq. 4.29 is computationally expensive, requiring a discretisation of the trajectory and the measurement space [28]. To reduce the computational effort, one can calculate the maximum instead of the integral. Specifically, we define the most likely contact frame ϕ_c^* and the most likely touch measurement z_{touch}^* as follows:

$$\phi_c^* = \arg\max_{\tilde{\phi}_c} [P(\phi_c | \mathbf{x}, \tau)]. \tag{4.30}$$

$$z_{touch}^* = \{z_P^*, z_n^*\} = \arg\max_{z_{touch}} [P(z_{touch} | \mathbf{x}, \phi_c^*, C)]. \tag{4.31}$$

So, the expected information gain for an action τ is approximated as in Eq. 4.32.

$$E[IG(\mathbf{x}, \tau)] \approx P(z_{touch}^*) IG(\mathbf{x}, z_{touch}^*). \tag{4.32}$$

In practice, the expected information gain of an action τ is approximated by the information gain obtained if the robot contacts the object in frame ϕ_c^*, multiplied by the probability of the most likely contact-point and normal-vector tuple. As in the examples presented in this thesis, if actions are generated with respect to the most likely pose of the object $\hat{\mathbf{x}}$, z_P^* corresponds to the center of f_k and z_n^* corresponds to n_k. Figure 4.29 illustrates τ, z_{touch}^*, ϕ_c^*.

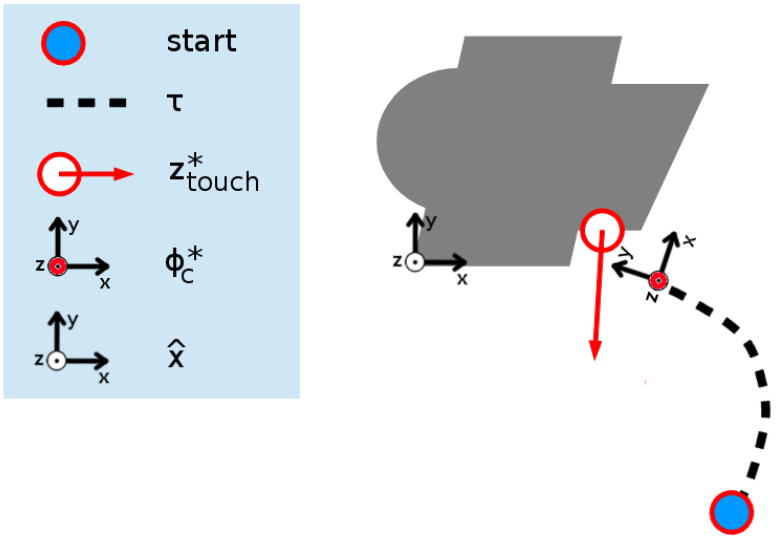

Figure 4.29: An action is comprised of a trajectory τ which is a collection of frames $\{\phi\}$ used as requirement for the controller. The most-likely contact frame of the end effector is ϕ^*_c, with z^*_{touch} being the most-likely contact measurement.

4.4.3 Information Gain Metrics for Action Selection

As presented in the previous Section, in active sensing, robot actions are selected so as to maximise the expected information gain. In a Bayesian framework, several metrics are available to measure such quantity. A good survey on this subject is proposed in [48]. Below, a set of suitable metrics is presented together with their characteristics with respect to touch-based active sensing. For the sake of clarity, they are grouped into two categories: those based on the covariance matrix and those based on the probability distribution.

- **Covariance-based metrics**: measuring the dispersion of a probability distribution with respect to its mean, the covariance matrix Σ is defined as:

$$\Sigma_{ij} = E[(x_i - \mu_i)(x_j - \mu_j)] \tag{4.33}$$

where x_i is the i-th state of $\mathbf{x}$, and $\mu_i = E[x_i]$ is the mean value of x_i. Active sensing aims at reducing the dispersion of the estimation, so minimising the covariance matrix corresponds to maximising the

information about the state. Hence, actions may be selected comparing their expected posterior covariance. However, reasoning about whole matrices is not a trivial task. Therefore, a number of covariance-based scalar metrics have been developed in literature, namely its determinant $|\Sigma|$, its trace $tr(\Sigma)$, its weighted trace $tr(W\Sigma)$ or its eigenvalues $\{\lambda(\Sigma)\}$. For our application, this corresponds to quantifying the information gain of a trajectory τ with associated most-likely measurement z^*_{touch} as the expected covariance metric of the posterior distribution over $\mathbf{x}$ given z^*_{touch}:

$$E[IG(\mathbf{x}, \tau)] \approx P(z^*_{touch}) f(P(\mathbf{x}|z^*_{touch})), \tag{4.34}$$

where f corresponds to one of the aforementioned covariance operators (determinant, trace, weighted trace or eigenvalues). In case of inference with particle filter, the covariance matrix can be extracted by the sample set [63].

- **Metrics based on the probability distribution**: The concept of probability **entropy** [60] was originally formulated in information theory, referring to the expected information carried by the value of the random variable x. Intuitively, it represents the amount of information required on the average to describe the random variable. Under the continuity convention that $0 \log 0 = 0$, it is defined as in Eq. 4.35.

$$H(P(x)) = E[-\log P(x)] = -\int_X P(x) \log P(x) dx. \tag{4.35}$$

Similarly to covariance-based metrics, entropy measures the uncertainty about state estimation by means of a scalar value and it represents an absolute measure that depends only on the distribution and which can be adopted to quantify the progress of the localisation. In addition, with respect to the covariance matrix, it is better suited for capturing the dispersion within single clusters of a multi-modal probability distribution [6]. Hence, in the context of active sensing, actions may be chosen so as to **minimise the probability entropy**, as proposed in [32, 4].

For our application, what we are interested in is the entropy of the probability on $\mathbf{x}$ conditioned on the measurement z^*_{touch}. Further to Eq. 4.32:

$$E[IG(\mathbf{x}, \tau)] \approx P(z^*_{touch}) H(P(\mathbf{x}|z^*_{touch})). \tag{4.36}$$

In practice, computing the integral in 4.35 may become computationally expensive in case of large state space and non-linear Bayesian filters.

Solutions [62, 6] have been proposed to calculate the entropy in case of a running particle filter, yet at high computational cost. Nikandrova *et al.* [52] adopted a weight-based approach to evaluate entropy of a Particle Filter treating the set of weights as a set of probability masses, thus ignoring their spatial distribution (Eq. 4.37). Formally, this does not represent an approximation of probability entropy and it may not be compared between different particle samples, *e.g.* after the resampling step. Hence, weight-based entropy is not an absolute metric of uncertainty. For these reasons, it will be noted as $\tilde{H}$ to distinguish it from H.

$$\tilde{H(\mathbf{x})} = -\sum_{i=1}^{N} w^{[i]} \log w^{[i]}. \tag{4.37}$$

However, $\tilde{H}$ was proven to be an effective metric to encode information gain when comparing actions leading to different weights for the same set of particles. The main advantage of this weight-based approach is its computational efficiency.

Alternatively, in the same work, the posterior distribution was approximated by a weighted kernel estimate formed from the collection of samples and weights of a running particle filter:

$$P(\mathbf{x}) \approx \sum_{i=1}^{N} w^{[i]} k(\mathbf{x} - x^{[i]}), \tag{4.38}$$

where the kernel $k(.)$ is a density function known as Parzen window [55], for instance taking the form of a multivariate Gaussian distribution. This provides means to extract a density function from a set of weighted particles. The expectation in Eq. 4.35 is approximated by the expectation over the weighted samples set $\{x^{[i]}\}$:

$$H(\mathbf{x}) \approx \sum_{j=1}^{N} w^{[j]} \log(\sum_{i=1}^{N} w^{[i]} k(x^{[j]} - x^{[i]})). \tag{4.39}$$

The entropy value computed in Eq. 4.39 is affected by the choice of the kernel function, which remains a parameter to be set by the robot programmer in literature examples. However, in the context of localisation applications, it represents an absolute approximation and can be used to measure information-gain progress through time with different particles sets. Nevertheless, its computation is significantly more intense than of the weighted-based entropy in Eq. 4.37.

In addition to probability entropy, the **Kullback-Leibler Divergence** (or Relative Entropy) [40] is a metric to quantify the difference between two probability distributions. Mathematically, it corresponds to the expected logarithmic difference between two probability distributions $p(x)$ and $q(x)$, as in Eq. 4.40.

$$D_{\mathrm{KL}}(p||q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx. \tag{4.40}$$

In general, the D_{KL} is not symmetric with respect to its members, thus, formally, it may not be considered a distance:

$$D_{\mathrm{KL}}(p||q) \neq D_{\mathrm{KL}}(q||p). \tag{4.41}$$

In a decision-making context, by definition, D_{KL} allows us to evaluate how much the posterior changes with respect to the prior after an action, quantifying such different through a scalar value. In our application, this corresponds to calculating D_{KL} of $P(\mathbf{x}_{t+1}|z^*_{touch})$ with respect to $P(\mathbf{x}_t)$:

$$E[IG(\mathbf{x}, \tau)] \approx P(z^*_{touch}) D_{\mathrm{KL}}\Big(P(\mathbf{x}_{t+1}|z^*_{touch})||P(\mathbf{x}_t)\Big). \tag{4.42}$$

However, with respect the entropy, it does not provide an absolute quantification of the uncertainty of the system, but rather a measurement of its evolution between two time steps. Hence, it can be used to compare the change in the probability distribution and **select the action with the highest expected divergence**. It is not suitable to quantify the progress of the localisation, unless a start and/or a desired final probability distribution are defined.

In general, covariance-based metrics for active-sensing decision making are effective tools to quantify information gain when the probability distribution over the state presents uni-modal behaviour, thus the mean value represents a good estimate of the actual state. They are computationally efficient, and allow the robot to extrapolate the direction of motion to acquire useful information [43]. Their invariance properties make them a suitable metric also to quantify the progress of the localisation in absolute terms. Instead, entropy and Kullback-Leibler divergence represent alternative ways to evaluate the information gain capturing the multi-modal nature of the posterior distribution, paying the price of a higher computational cost.

4.4.4 Information-Gain Metrics Performance for V-block Localisation

A set of trials have been carried out to compare the performance of the information metrics presented in the previous subsection. The task is to localise obj_{VB} with initial uncertainty of [0.1m 0.1m 1rad]. As in the previous section, a particle filter inference model is adopted. Once 80% of the probability mass on the position of the target is concentrated within a radius of 5mm, the end-effector moves down aiming at the target feature defined on the object. As for the action-selection scheme, the time horizon is set to $h = 1$, *i.e.* the action is chosen as to maximise the immediate reward. In decision-making literature, this approach is also referred to as *greedy*.
Decision-making strategies based on **random**, **covariance-determinant minimisation** and **entropy minimisation** action selection are compared. Entropy is approximated as in Eq. 4.37. The latter was preferred to a Kullback-Leibler divergence strategy for its reduced computational cost, since at time t it does not require to store the probability distribution at time $t-1$. The objective of the experiment is to evaluate the performance of a fully motion-oriented, random strategy to more information-driven approaches which require a longer time to reason about the 11 actions depicted in Figure 4.24. Tests were carried out keeping the obj_{VB} on the same location, performing the same initial sweep (as depicted in Figure 4.30) and approximately the same initial conditions. Ten trials were repeated for each decision-making strategy.

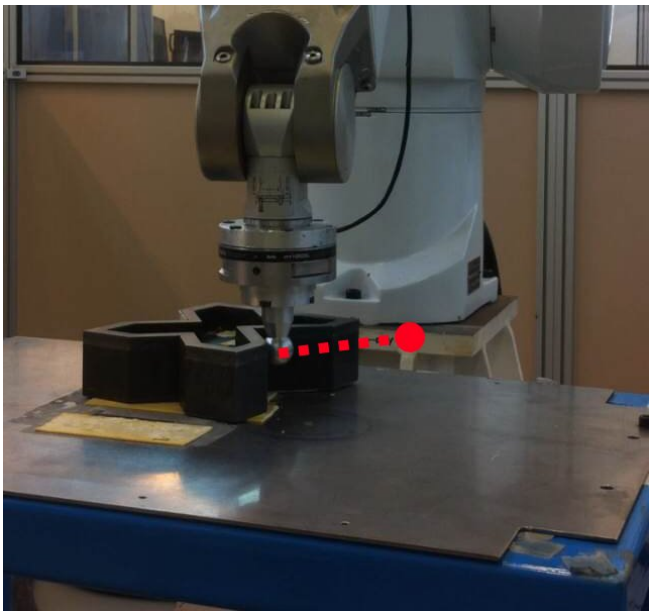

Figure 4.30: V-block localisation with initial uncertainty [0.1m 0.1m 1rad]: the same initial measurement was repeated throughout the tests to compare random action selection to covariance-based and entropy minimisation strategies starting from the same information conditions.

The time required to complete the task and the error of the localisation have been recorded comparing the final position of the end effector with that of the target. The latter was obtained manually moving the robot on the spot, so the error was measured against a ground truth that depends on human accuracy, which we estimate with a tolerance error on the order of 2mm.
The final errors obtained with the three action-selection strategies are shown in Figure 4.31. Two outliers have been recorded due to contacts on a vertex of the v-block. This fed a misleading measurement to the particle filter, so the final target estimation was biased, as described in Section 4.1. Overall, the three strategies present comparable results in terms of estimation reliability.

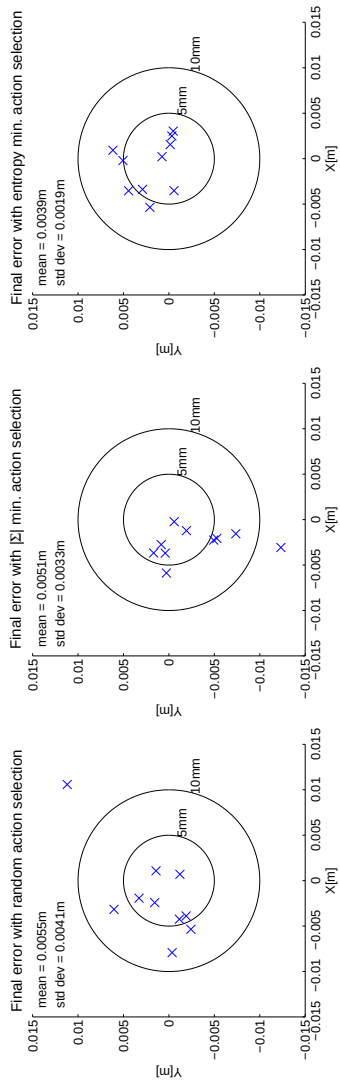

Figure 4.31: V-block localisation with initial uncertainty [0.1m 0.1m 1rad]: final localisation error measured with three different decision-making strategies calculated with respect to a human-defined ground truth. The mean and standard deviation of the norm of the error is reported in the graph.

The average time required to achieve the localisation task ($\bar{t}_{\text{tot}}$) and to establish a second contact ($\bar{t}_{\text{2nd}}$) are reported for the three decision-making strategies in Table 4.3. In this application, a random selection outperforms the two more information-gathering strategies in terms of total execution time. In particular, it generates a speedy second contact after the first one depicted in Figure 4.30, about 30% faster than the other two methods. In particular, the second contact proves to be most informative action in terms of rate of uncertainty reduction. As for the number of required sensing actions, entropy minimisation needs approximately two actions less than the other two methods. This corresponds to selecting more informative actions, paying the price of a higher computational time. Figure 4.32 compares the localisation progress with the three strategies measured through the maximum eigenvalue of the covariance matrix. In Figure 4.33, the localisation progress in terms of probability mass within the 5mm tolerance is plotted for the three decision-making strategies.

Strategy	$\bar{t}_{\text{2nd}}$	$\bar{t}_{\text{tot}}$	avg Actions
random	**8.4s**	**54.7s**	7.9
$\lvert\Sigma\rvert$ minimisation	12.1s	99.9s	7.9
H minimisation	12.0s	61.0s	**5.6**

Table 4.3: v-block localisation with initial uncertainty [0.1m 0.1m 1rad]: comparison of performance with different action-selection strategies. The table reports the average elapsed time before the second contact $\bar{t}_{\text{2nd}}$, the total time to execute the task and the average number of actions required.

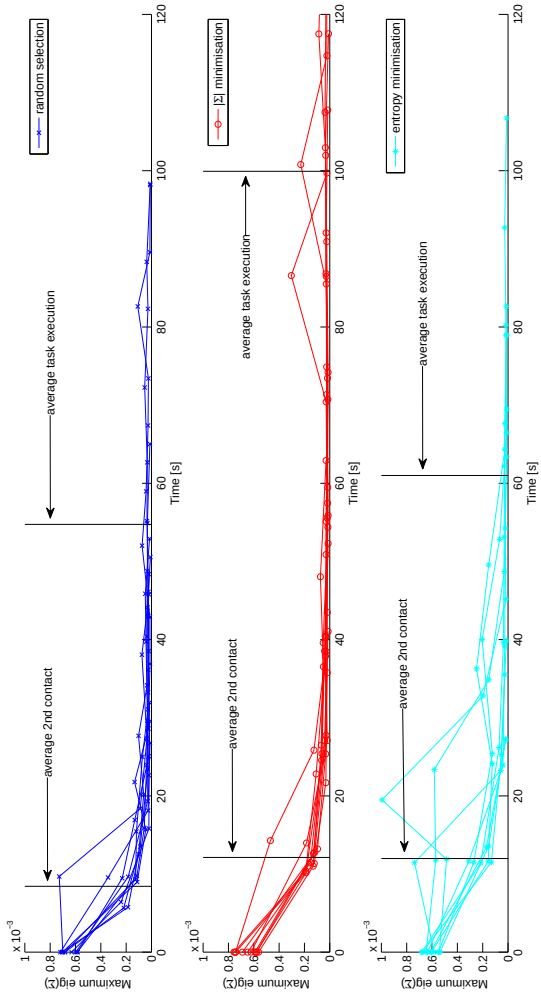

Figure 4.32: V-block localisation with initial uncertainty [0.1m 0.1m 1rad]: comparison of localisation progress between different decision-making strategies: random selection, covariance-determinant minimisation and entropy minimisation. The maximum eigenvalue of the covariance matrix is plotted against running time. Vertical lines illustrate the average time required to establish a second contact which is beneficial to significantly reduce the uncertainty at the beginning of the localisation.

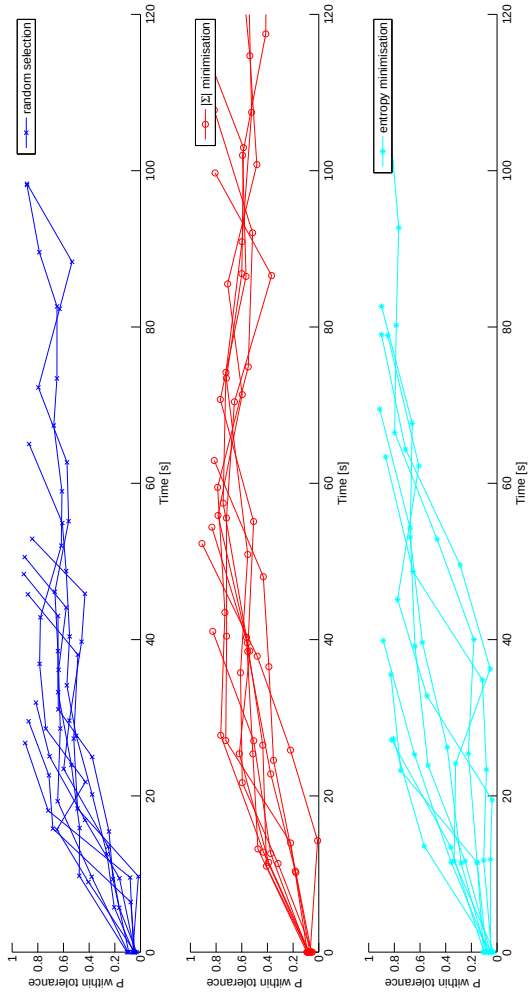

Figure 4.33: V-block localisation with initial uncertainty [0.1m 0.1m 1rad]: comparison of localisation progress between different decision-making strategies: random selection, covariance-determinant minimisation and entropy minimisation. The probability mass within a 5mm tolerance is plotted against running time.

In conclusion, this Section has presented different information gain metrics used as reward function to select which action to perform. In the reported case study with initial uncertainty of [0.1m 0.1m 1rad], random action selection allowed the robot to localise a v-block faster than using entropy or covariance-based metrics. This is due to the heavy computation required by the simulation of each option in the action space. However, an information-gathering strategy based on entropy minimisation guarantees convergence after a smaller number of actions, yet requiring a longer time to evaluate the executable actions. These results motivate us to further investigate means to perform the same task adjusting the complexity of the decision making as the localisation progresses. In the following section, the *act-reason* algorithm presented in Section 3.4 is adopted to vary the action space at run time, taking advantage of a random initial action selection and a more thoughtful decision making based on entropy minimisation.

4.5 Reducing Global Execution Time in Localisation Tasks with Act-reason

In Section 4.4.3, strategies to select actions based on different information-gain metrics have been compared to localise the v-block object. Measuring task progress as the reduction of uncertainty, we observe a better performance of a random decision making especially at the beginning of the localisation, while an entropy minimisation strategy generates more informative actions yet taking longer for their evaluation. Intuitively, at the beginning of the localisation, very little information is available and *any* contact is likely to reduce the uncertainty significantly. At this stage, one would wish to follow a random selection to obtain fast contacts. Instead, when more information is available, spending more time reasoning to find *the* best action through entropy minimisation tends to pay off.

4.5.1 Act-reason to Reduce Execution Time

In order to exploit both the benefits of an **initial random strategy** and the **informative actions** obtained with an entropy-based selection, we adopt the *act-reason* algorithm introduced in Chapter 3 to vary the allocated time to reason and execute depending on the current uncertainty, as in Eq. 4.43. Specifically, we set the allocated time t_{alloc} to vary between the average time required by a random action $\hat{t}_{rand}$ and that of a full-resolution entropy minimisation $\hat{t}_{ent}$, considering both computation and execution. More precisely, this variation is made linear in the progress metric Π:

$$t_{alloc} = \hat{t}_{rand} + \Pi\,(\hat{t}_{ent} - \hat{t}_{rand}). \tag{4.43}$$

In our implementation, Π is defined as the ratio between the current probability mass within the 5mm tolerance and the desired value to call the localisation done.

$$\Pi = \frac{P(\mathbf{x} \in tol)}{P(\mathbf{x} \in tol)_{final}} \tag{4.44}$$

Intuitively, this corresponds to adjusting the robot behaviour from fully random when the uncertainty is high, to fully info-gathering when the uncertainty is low and more informative actions improve convergence, as illustrated in Figure 4.34.

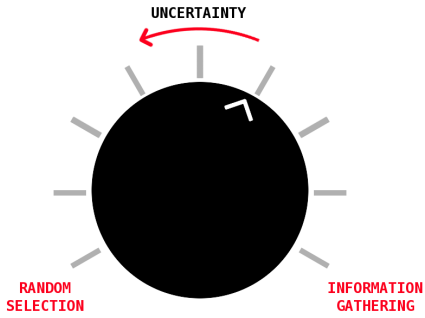

Figure 4.34: Graphical illustration of the intuition behind the *act-reason* algorithm which allows the robot programmer to vary the resources allocated to choose and execute the next action.

Figure 4.35 and Figure 4.36 present results in terms of information gain vs time during a v-block localisation with initial uncertainty of [0.1m 0.1m 1rad] and [0.1m 0.2m πrad]. In both cases, applying the *act-reason* algorithm to vary t_{alloc} as in Eq. 4.43, a faster second contact and a lower execution time is recorded. In Table 4.4and 4.5, the the time to second contact and the task-execution time and are reported.

Uncertainty	*act-reason* $\bar{t}_{2\text{nd}}$	H min. $\bar{t}_{2\text{nd}}$	random $\bar{t}_{2\text{nd}}$
0.1m 0.1m 1rad	8.7s	12.0s	**8.4s**
0.1m 0.2m rad	**8.0s**	10.4s	-

Table 4.4: v-block localisation: comparison of time to second contact with and without *act-reason* algorithm.

Uncertainty	*act-reason* $\bar{t}_{\text{tot}}$	H min. $\bar{t}_{\text{tot}}$	random $\bar{t}_{\text{tot}}$
0.1m 0.1m 1rad	**40.1s**	61.0s	54.0s
0.1m 0.2m πrad	**52.1s**	55.8s	-

Table 4.5: v-block localisation: comparison of task-execution time with and without *act-reason* algorithm.

Figure 4.35: V-block localisation with initial uncertainty [0.1m 0.1m 1rad]. Progress vs time with fixed action space (cyan) and variable action space using the *act-reason* algorithm (green).

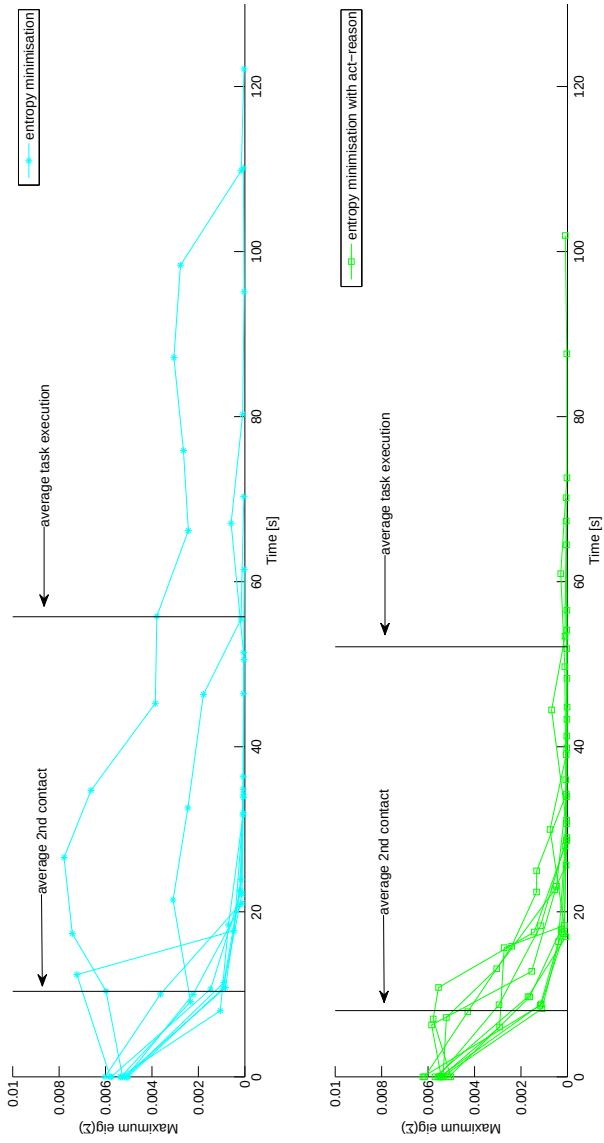

Figure 4.36: V-block localisation with initial uncertainty [0.1m 0.2m πrad]. Progress vs time with fixed action space (cyan) and variable action space using the *act-reason* algorithm (green).

4.5.2 Analysis

This Section has presented two applications where *act-reason* has enabled the robot to improve its performance in a 3DOF localisation task in terms of total execution time. Specifically, the allocated time for each sensing action was set proportional to the progress of the localisation measured as the probability mass within a specified tolerance, setting the robot behaviour from explorative to speculative depending on the available information about the state. Compared to a classic entropy minimisation action-selection, a 35% improvement in terms of execution time has been achieved in applications with initial uncertainty of [0.1m 0.1m 1rad]. In case of higher initial uncertainty, the current time-allocation scheme has achieved a time gain of 7%. Indeed, the stochastic position of the particles may affect, to some extent, the calculation of Eq. 4.37, thus the rate of convergence of the localisation. However, in both the presented cases, adjusting t_{alloc} was beneficial and further research may focus on improving its relationship with respect to the current level of uncertainty.

4.6 Conclusion and Remarks

This Chapter has illustrated the measurement setup adopted in the robot experiments, which is comprised of a contact detector that acquires information when the robot moves in free space and a contact-point and normal-vector estimator collecting information by physical interaction. Trials carried out to estimate the precision of the measurement apparatus have been presented.
In order to assess the **scalability** of the DOF Decoupling Methodology presented in Chapter 3, a 3DOF localisation of an object on top of a table has been simulated with initial uncertainty up to [1m 1m πrad]. A concrete implementation of the contact-detection measurement model has been presented, explicitly taking into account the misalignment between the adopted geometric model and the actual object. The running time of the whole task was recorded with and without decoupling. In the former case, the time required to accomplish the task was proportional to the square of the side of the table, whereas in the latter case it scaled with its cube.
Three objects of increasing complexity have been localised by the robot with initial uncertainty of [0.1m 0.1m 1rad]. To the best of our knowledge, we applied touch-based localisation to objects whose mesh complexity (up to $\approx$1000 vertices) is about an order of magnitude higher than those presented in literature [57] [28] [32]. The recorded mean and standard deviation estimation error are [μ =0.0050m, σ =0.0025m] [μ =0.0043m, σ =0.0034m] [μ =0.0073m, σ =0.0048m].
Different action-selection schemes have been tested on a case-study object localisation, namely: random, covariance-based and entropy-minimisation

action-selection. For the studied example, a random selection proved more effective in terms of total execution time, whereas entropy-minimisation allowed the robot to localise the object with less actions.
Building upon these results, the *act-reason* algorithm introduced in Chapter 3 has allowed the robot **to reduce the execution time** adjusting its behaviour from random to info-gathering setting the allocated time for the next action with respect to the current uncertainty. Two case studies with different initial uncertainty have been presented, reporting a reduction of running time of 35% and 7%. To the best of our knowledge, this represents the **first example of action space with variable complexity** in a touch-based localisation application.

Chapter 5

General Conclusion

This chapter summarises the scientific results obtained in this thesis and makes concrete suggestions for future extensions.

5.1 Scientific Results

- An **experimental study** with a group of human beings facing a robot-like object localisation using force sensing has been carried out. Filmed throughout the test, the subjects showed common patterns such as following the same chain of actions and reducing the uncertainty over different degrees of freedom in sequential fashion. Hypothesised reasoning leading to such behaviour inspired the formulation of the DOF Decoupling Methodology.
- A **model-level description of active sensing task** was presented based on five modelling concepts: configuration space, information space, action space, inference scheme and action-selection scheme. The proposed model was obtained generalising from prior relevant examples, and concrete instances from the literature have been provided.
- **DOF Decoupling** was introduced as a **methodology** to describe localisation applications as a task graph composed of subtasks. This formulation allows the robot programmer to minimise the complexity of the configuration space in each subtask, thus reducing the cost of planning actions. The effectiveness of the methodology was **validated** with a 3DOF force-based object localisation application. By comparison with task execution without decoupling, significant reduction in the complexity of the task has been obtained. A measurable improvement on the **scalability** of the problem in case of high initial uncertainty has been achieved.

- The *act-reason* algorithm was introduced and implemented as **action-selection scheme** to choose the next best action to perform reasoning over a variable action space. Focusing on the reduction of the total time required for localising the object, it trades off computation, execution and information gain. With respect to the state of the art, this scheme allows the robot programmer to have a deterministic influence on the time spent for each of the robot action, taking into account the time for compution, evaluation and execution. Case-study examples are presented to show how *act-reason* can be used to improve the time efficiency in touch-based localisation tasks with respect to naive action-selection schemes.
- Robot localisation applications were performed with objects up to **industrial-relevance complexity** using a Staubli RX90 equipped with a spherical end-effector coupled with a force-torque.

5.2 Discussion and Future Work

This thesis has identified four major fields of research involved in active sensing applications: probabilistic state estimation, geometric modelling, action definition and action selection. While they all affect the complexity, reliability and time-efficiency of the localisation task, they present different levels of maturity from a scientific and engineering point of view. Specifically, parametric and non-parametric Bayesian estimation algorithms have been developed for the past 60 years, reaching significant results in terms of formalisation and implementation. Several CAD and meshing software packages are available on the market, providing a rather wide range of tools for geometrical object modelling. Instead, action definition and selection remain the most unexplored areas in this context. Software packages such as Graspit! [49] provide useful means to generate grasp configurations and reaching actions for hand tools. However, for each hand-object tuple the set of possible grasping configuration is potentially enormous, thus a number of assumptions are made to reduce the size of the so-called grasp wrench space (GSW), *e.g.* by using a discretisation of the achievable friction forces on each contact point. In this work, the *act-reason* algorithm has introduced a decision-making framework which provides the robot programmer with a tool to constrain the time spent to evaluate and execute the next action making the action space cardinality flexible. In this context, prioritising task-relevant actions at run time is crucial since this allows the robot to reason starting from the actions there are most likely to be useful. More specifically, we foresee the industrial potential of a **toolbox** able to:

- generate sensing actions taking into account the geometry and dynamic properties of the object together with those of the robot;

- prioritise potentially useful actions at run time, to feed the most task-oriented motion primitives to the decision-making algorithm. Actions should be (pre-)evaluated based on their kinematic feasibility, execution time and probability of success. Intuitively, the robot should not spend time reasoning about difficult or useless actions.

From a scientific point of view, we identify the following research streams to enhance the results presented in this thesis.

1. Building upon the test described in Chapter 2, further experiments should **deepen the study on human behaviour in force-based localisation applications**, for instance by limiting the time to execute the whole task and requiring different levels of final uncertainty. One could also apply some motion capture routine in order to extrapolate action from human demonstration.

2. This thesis has shown how task graph, policies and action space may be used to enhance localisation applications focusing on minimising the total execution time. However, from an artificial intelligence perspective, these are all elements that can be **learnt**.

   Applying an empirical DOF Decoupling to define the task graph, significant complexity reduction was achieved on a 3DOF force-based localisation application. Extending these results by defining the task graph out of a **constrained optimisation** would represent a remarkable extension of our findings. In this regard, the Distributional Clauses Particle Filter framework [53] might be adopted to encode **topological relations between different objects** such as {*inside, on top, beside*} in order to define the structure of the task graph. This may allow the robot to prioritise which objects to localise first in a complex scene such as the one proposed in the human experiment.

3. This thesis has focused on localisation tasks of a single known object. Possible extensions may look ahead and study the problem of **localising multiple objects**. This will result in a higher complexity of the configuration space, due to extra DOFs and the additional probability distribution about which item is being sensed by the robot. Hence, applying the DOF Decoupling Methoology would be beneficial to fight such complexity. Moreover, in order to **localise objects whose geometric model is (partially) unknown**, one could take advantage of the vast literature on Simultaneous Localisation and Mapping (SLAM) [63] which treats the similar problem of building a map while localising a mobile robot within it. However, standard SLAM applications are comprised of low-dimensional spaces (*e.g.* 2.5D). Therefore, the DOF Decoupling

Methodology is expected to bring in a reduction of complexity to increase the on-line feasibility of the task.

4. **Robot and environment constraints** should be also taken into account during action selection. Specifically, a **kinematic feasibility** module can be added in order to prune those actions which fall into low dexterity spaces. In the presented examples, the reward function in Eq. 3.1 took into account only the information gain. However, one can apply the same action-selection model taking into account negative metrics such as the risk of unexpected collisions or the cost of motion. It seems sensible to use time as unit of measurement of such reward. Specifically: information gain (hence uncertainty) can be mapped to the estimated seconds required to complete the localisation; the risk of collision can be mapped to the time to fix the damage, and the time of motion is already homogeneous with respect to time.

5. Evidence shows how the complexity of the active-sensing task is also influenced by that of the geometric model used to represent the object. Future work should investigate the impact of geometric model fineness on task complexity, possibly extending the five primitive concepts belonging to the Active Sensing Model.

6. The current implementation of *act-reason* requires the robot to not move while making the decision about where-to-sense-next. Intuitively, while the robot stays still it is losing information, or in the best-case scenario, it is not acquiring any useful measurement. Hence, future work should focus on **making the decision while moving**, without obliging the robot to freeze if not necessary. Moreover, as *act-reason* makes the number of actions to consider variable, also the time horizon h should vary too.

7. The DOF Decoupling Methodology, the Active Sensing Task Model and the *act-reason* algorithm may be extended to **multi-sensor systems**, allowing the robot to make decisions about which sensor(s) to use in a task-oriented fashion. From an industrial point of view, this would make the scene-model calibration more robust as different sensors increase the set of scenarios in which this operation becomes possible.

8. While humanoid control has seen a rather intense development in recent years (see for example [23] [44]), and systems with multiple tactile modules have been successfully integrated on robot manipulators [50], **whole-body touch-based active sensing** remains a rather unexplored field of research. In particular, such applications present a highly-complex action space, where the robot interacts with the environment possibly carrying out several tasks at the same time, *e.g.* walking, touching and

scanning the surrounding area with a laser. Hence, the application of DOF-Decoupling is expected to reduce the complexity of the active-sensing application, improving its scalability and online feasibility.

Chapter 6

Other Contributions

6.1 Integration of the TAO Controller Interface in Orocos

The robot experiments presented in this thesis have been implemented on a Staubli RX90 robot controlled by the TAO (Téléopération Assistée par Ordinateur) software architecture developed at CEA. Running on a VXWorks real-time machine, it features a C-language interface named *taolib* to programme robot actions. Mainly conceived for use on Windows OS, *taolib* was compiled for Linux and encapsulated into an Orocos[7] component. Installation and tutorials on how to use the interface on a Linux OS were documented on the CEA wiki page.

6.2 Localisation of the Stop Assembly Button

In the framework of the EU-funded Rosetta Project, a stop-button localisation was implemented in collaboration with other PhD researchers at KULeuven. The task was to securely lock the stop button to the edged corner of the table. This was performed decoupling the task as follows:

- subtask 1: constrain the object to the edge
- subtask 2: push the object along the edge up to the desired corner

An internal report was produced and code was released for dissemination among project partners.

Chapter 7

Appendix

This Appendix chapter presents implementation details and derivations meant to provide the reader with a sound understanding of the physical and theoretical tools adopted throughout the thesis.
Due to the extensive naming required in this Chapter, symbols are defined inline and the adopted nomenclature might not match the one used in the rest of the thesis.

7.1 Contact Point and Normal Vector Estimation With Pseudo Contact Point

A force-torque sensor coupled with a spherical end effector allows the robot to estimate the contact point and normal vector of the touched face. First introduced by Kitagaki in [38], under the assumption of no friction between the pin and the surface, the model adopted in this work is here presented.
The following equations refer to the naming conventions summarised in Figure 7.1. Specifically, the contact point is noted as $\mathbf{M}$, the origin is $\mathbf{O}$, and $\mathbf{F} = \{Fx, Fy, Fz\}$ and $\mathbf{T} = \{Tx, Ty, Tz\}$ represent the force and torque vectors measured by the sensor in $\mathbf{O}$. The coordinate of the pseudo-contact point $\mathbf{H}$ can be worked out as:

$$\mathbf{H} = \begin{cases} H_x = \frac{Tz\,Fy - Ty\,Fz}{F^2} \\ H_y = \frac{Tx\,Fz - Tz\,Fx}{F^2} \\ H_z = \frac{Ty\,Fx - Tx\,Fy}{F^2}. \end{cases} \tag{7.1}$$

The contact point $\mathbf{M}$ corresponds then to the intersection between the straight line $\mathbf{H} + k\,\mathbf{F}$ and the end effector sphere centred in $\mathbf{C}$:

$$\begin{cases} M_x = H_x + k\,Fx \\ M_y = H_y + k\,Fy \\ M_z = H_z + k\,Fz \\ (M_x - C_x)^2 + (M_y - C_y)^2 + (M_z - C_z)^2 = R^2. \end{cases} \tag{7.2}$$

Where R is the radius of the sphere. Since the straight line intersects the sphere in two points, namely $\mathbf{M}_1$ and $\mathbf{M}_2$, two solutions are found for the system in Eq. 7.2:

$$k_{1,2} = \frac{-(Fx\,\Delta_x + Fy\,\Delta_y + Fz\,\Delta_z) \pm \sqrt{\Delta}}{F^2}. \tag{7.3}$$

$$\mathbf{M}_1 = \mathbf{H} + k_1\,\mathbf{F}. \tag{7.4}$$

$$\mathbf{M}_2 = \mathbf{H} + k_2\,\mathbf{F}. \tag{7.5}$$

Where:

$$\begin{cases} \Delta_x = H_x - C_x \\ \Delta_y = H_y - C_y \\ \Delta_z = H_z - C_z \end{cases} \tag{7.6}$$

and

$$\Delta = 4(Fx\,\Delta_x + Fy\,\Delta_y + Fz\,\Delta_z) - 4F^2(\Delta_x^2 + \Delta_y^2 + \Delta_z^2 - R^2). \tag{7.7}$$

The selection between $\mathbf{M}_1$ and $\mathbf{M}_2$ is done selecting the one that guarantees $\mathbf{CM} \cdot F < 0$, hence a repulsive force on the pin. Consequently, the normal vector is worked out as:

$$\mathbf{N} = \frac{\mathbf{CM}}{||\mathbf{CM}||}. \tag{7.8}$$

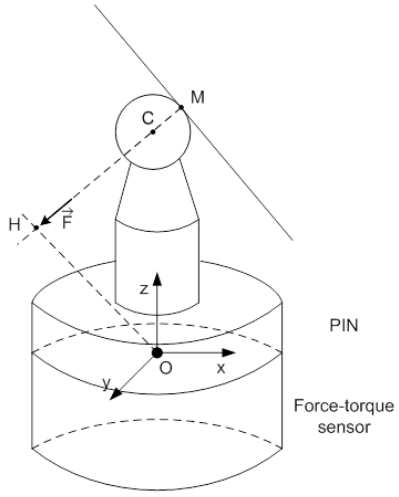

Figure 7.1: Pseudo-contact technique to estimate contact point and normal vector using a spherical end effector coupled with a force-torque sensor.

7.2 Probability Theory

Probability theory is the branch of mathematics that deals with stochastic events, providing a suitable framework for reasoning under uncertainty. This Section presents an overview on probability theory and its applications to recursive state estimation and decision making under uncertainty in robotics. The aim is to provide the reader with a sound understanding of the basic notions and terminology which are adopted throughout this thesis.

7.2.1 Random Variables

In probability theory, a random variable can take multiple values which represent the results of the non-deterministic outcome of an experiment. It does so according to specific probabilistic laws. In particular, we distinguish:

- **discrete random variables** which map the events of a random process to a countable set of distinct values;
- **continuous random variables** which map the events of a random process to an uncountable set of distinct values.

7.2.2 Probability Distributions

Probability distributions characterise a random variable by assigning probabilities to the values it can take. In case of a discrete random variable, this distribution is called *probability mass function* (PMF), and represents the probability that the random variable X takes a value x:

$$P(X = x) \tag{7.9}$$

By definition, a PMF is non-negative and must sum to one:

$$P(X = x) \geq 0, \tag{7.10}$$

$$\sum_{x} P(X = x) = 1. \tag{7.11}$$

For the sake of brevity, the notation $P(x)$ for $P(X = x)$ is commonly used in literature and is adopted throughout this thesis. In case of a continuous random variable X, the *probability density function* (PDF) represents the probability that X occurs at a given place in the uncountable output set D of the continuous random variable. The integral of the PDF within a region $[a, b] \in D$ is the probability of X falling into this region:

$$P(x \in [a, b]) = \int_{a}^{b} P(x)dx. \tag{7.12}$$

As for the discrete case, a probability density function is defined non-negative:

$$P(x) \geq 0. \tag{7.13}$$

And its integral sums to 1:

$$\int_{D} P(x)dx = 1. \tag{7.14}$$

The *joint probability* of two random variables X and Y represents the probability that X takes a certain value x and that Y takes a certain value y. Formally, the joint probability is expressed by $P(x, y) = P(X = x \text{ and } Y = y)$.
Two random variables X and Y are *independent* whenever

$$P(x, y) = P(x)P(y). \tag{7.15}$$

7.2.3 Marginalisation, Independence and Bayes Rule

The *conditional probability* expresses the probability of X taking a certain value x given that $Y = y$ and is defined as:

$$P(x|y) = \frac{P(x,y)}{P(y)}. \tag{7.16}$$

Given the joint density $P(x|y)$, the *marginal probabilities* $P(x)$ and $P(y)$ can be recovered by applying the *Theorem of Total Probability*. In the discrete case:

$$P(x) = \sum_y P(x,y) = \sum_y P(x|y)P(y) \tag{7.17}$$

and

$$P(y) = \sum_x P(x,y) = \sum_x P(y|x)P(x). \tag{7.18}$$

In the continuous case:

$$P(x) = \int_y P(x,y)dy = \int_y P(x|y)P(y)dy \tag{7.19}$$

and

$$P(y) = \int_x P(x,y)dx = \int_x P(y|x)P(x)dx. \tag{7.20}$$

The Bayes' Theorem provides an expression of the conditional probability $P(x|y)$ in terms of $P(y|x)$.

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}. \tag{7.21}$$

7.3 Bayesian Inference

In this section, an overview on Bayesian inference and the most common algorithms for recursive state estimation adopted in robotics are presented. Throughout this thesis:

- the state at time t will be denoted by x_t
- the evolution of x_t through time is assumed to be a Markov chain, *i.e.* the next state x_{t+1} depends only on the current state x_t and the control u_{t+1} and not on the sequence of states, controls and measurements that preceded it, namely: $x_{0:t}, u_{0:t}, z_{0:t}$ (Markov assumption).

In mathematical terms:

$$P(x_{t+1}|x_{0:t}, u_{0:t+1}, z_{0:t+1}) = P(x_{t+1}|x_t, u_{t+1}). \tag{7.22}$$

7.3.1 Frequentist and Bayesian Inference

Probability for state estimation may be interpreted differently if following a *frequentist* or a *Bayesian* approach. The former approach considers probabilities as the long run frequencies with which events occur. Therefore, a frequentist estimator will estimate a state x by maximising its data likelihood function $P(z|x)$, where z are the measurement data. Instead, a Bayesian estimator will estimate x by maximising the posterior distribution $P(x|z)$ exploiting the Bayes theorem which allows to take into account both the data likelihood $P(z|x)$ and the prior $P(x)$. Bayesian inference is widely adopted in robotics applications, including this thesis, due to its capability to reason among world evidence explicitly accounting for prior information about the state and the uncertainty affecting the measurement.

7.3.2 Bayes Filter

The Bayes Filter is the most generic algorithm for iteratively estimating a state x_t modelled as an unobserved Markov process, given the measurement set $\{z_{1:t}\}$ which are the observed states of a hidden Markov model (HMM) at time steps $1, 2, ...t$. The target distribution to estimate is the posterior probability $P(x_t|z_{1:t})$. The Bayes filter is initialised with a prior distribution $P(x_0)$. Each iteration of the algorithm is comprised of two steps: *prediction* and *correction* (or update). In the prediction step, the new state is predicted using the prior information and the motion model $P(x_t|x_{t-1}, u_t)$, which models the state transition from time $t-1$ to time t given control u_t:

$$P(x_t|z_{t-1}, u_{t-1}) = \int P(x_t|x_{t-1}, u_t)P(x_{t-1}|z_{t-1})dx_{t-1}. \tag{7.23}$$

Then, the posterior probability is updated incorporating the measurement probability $P(z_t|x_t)$:

$$P(x_t|z_t) = \frac{P(z_t|x_t)P(x_t|z_{t-1}, u_{t-1})}{P(z_t|z_{t-1})}. \tag{7.24}$$

A single iteration of the Bayes Filter is presented in Algorithm 3. In practice, state estimation is initialised with a prior distribution $P(x_0)$, recursively running the algorithm for each new control u_t and measurement z_t. The η factor is defined such that $P(x_t|z_t)$ sums up to one.

7.3.3 Kalman Filter

The Kalman Filter [35] is probably the most common implementation of the Bayes Filter for recursive state estimation. Specifically, it is a parametric optimal

Algorithm 3 Bayes Filter

1: Initialisation $(P(x_{t-1}|z_{t-1}), u_t, z_t)$
2: **for** all x_t **do**
3: $P(x_t|z_{t-1}) = \int P(x_t|x_{t-1}, u_t)P(x_{t-1}|z_{t-1})\mathrm{d}x_{t-1}$ ▷ prediction
4: $P(x_t|z_t) = \eta P(x_t|z_{t-1})P(z_t|x_t)$ ▷ update
5: **end for**

linear estimator. Firstly introduced in 1958 for rocket trajectory tracking, since then it has been adopted in a large set of scientific disciplines. A number of extensions and generalisations have been proposed over the years, such as the Extended Kalman Filter [25] (EFK) and the Unscented Kalman Filter [67] (UKF). In this thesis, we present the original version of the algorithm. Detailed descriptions of Kalman filters, including its extensions, are available in literature, for instance [63].
Under the assumption of a linear dynamic system and Gaussian process and measurement noises (Eq. 7.3.3), the Kalman Filter (KF) provides an efficient algorithm to recursively estimate the state x_t.

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t + \delta_t \\ z_t &= C_t x_t + \epsilon_t. \end{aligned} \tag{7.25}$$

δ_t and ϵ_t are the process and measurement noises, which are both modelled with zero mean and with covariance matrices Φ_t and Γ_t respectively. Under these conditions, the conditional probability $P(x_t|z_t)$ is also Gaussian and is fully characterised by its mean μ and covariance Σ. Moreover, the KF is optimal as it minimises the mean square error of the estimated parameters. The matrices A_t, B_t, C_t, Φ_t and Γ_t are known and may be time-dependent. The filter is initialised by a prior distribution on x_0 represented by its mean μ_0 and covariance Σ_0.
The prediction probability of Eq. 7.3.2 can be calculated in closed form as a Gaussian distribution:

$$P(x_t|z_{t-1}, u_{t-1}) = \mathcal{N}(\bar{\mu}_t, \bar{\Sigma}_t) \tag{7.26}$$

with predicted mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$:

$$\bar{\mu}_t = A_t \bar{\mu}_{t-1} \tag{7.27}$$

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + \Phi_t. \tag{7.28}$$

The updated posterior is also a Gaussian distribution:

$$P(x_t|z_t) = \mathcal{N}(\mu_t, \Sigma_t), \tag{7.29}$$

with mean μ_t and covariance Σ_t:

$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \tag{7.30}$$

$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t. \tag{7.31}$$

K_t is the Kalman gain and specifies how much the new measurement information should be incorporated in calculating the new estimate:

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t + \Gamma_t)^{-1}. \tag{7.32}$$

Alg. 6 presents an iteration of the Kalman Filter in pseudo code.

Algorithm 4 Kalman Filter

1: Initialisation(μ_0, Σ_0)
2: $\bar{\mu}_t = A_t\bar{\mu}_{t-1}$ ▷ Predicted mean
3: $\bar{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + \Phi_t.$ ▷ Predicted covariance
4: $K_t = \bar{\Sigma}_t C_t^T (C_t\bar{\Sigma}_t C_t + \Gamma_t)^{-1}$ ▷ Kalman gain
5: $\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)$ ▷ Estimated mean
6: $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$ ▷ Estimated covariance

Intuitively, the Kalman Filter represents an efficient state estimator which provides optimal solutions for a simplified problem. In practice, the hypothesis of linear system may not always hold. To cope with this issue, the EKF and the UKF have been proposed to relax this assumption. They are approximated Bayes estimators as the found solution is not optimal due to the linearisation.

7.3.4 Particle Filter

The Particle Filter (PF) is a non-parametric recursive Bayes estimator which represents the conditional posterior $P(x_t|z_t)$ via a Monte-Carlo distribution. While it provides only an approximated solution, it is able to represent arbitrary distributions, making no assumption on the type of transition and measurement model.
The main idea is to approximate the posterior distribution by a set of weighted samples χ_t, named *particles*, drawn from it:

$$P(x_t|z_t) \approx < x_t^{[i]}, w_t^{[i]} > . \tag{7.33}$$

Each particle $x_t^{[i]}$ represents a concrete instance of the state x_t and is sampled from a so-called proposal distribution $\lambda(x)$. In case this corresponds to the transition model $P(x_t|x_{t-1}, u_t)$, the importance weights $\{w_t^{[i]}\}$ are proportional to the likelihood function $P(z|x)$. Intuitively, they encode the offset between

the proposal distribution, used for sampling, and the target distribution which corresponds to the posterior:

$$w_t^{[i]} = \frac{target}{proposal} \propto P(z_t|x_t^{[i]}). \tag{7.34}$$

An implementation of the Particle Filter is presented in Alg. 5 in pseudo-code. The number of particles n is a design parameter which affects the precision of the filter and can variate over time.

Algorithm 5 Particle Filter

```
Initialisation(χ_0, u_0, z_0)
for i = 1 : n do
    sample x_t^[i] ~ P(x_t|u_t, x_{t-1}^[i])        ▷ Draw from transition model
    w_t^[i] = P(z_t|x_t^[i])                        ▷ Update weight
    add x_t^[i] to χ̄_t
end for
for i = 1 : n do
    draw m from {x_t^[m], w_t^[m]}                  ▷ Resampling step
    add x_t^[m] to χ_t
end for
for i = 1 : n do
    w̃_t^[i] = w_t^[i] / Σ_0^N w_t^[i]               ▷ Normalisation
end for
```

7.4 Decision Making

This section presents two decision-making frameworks to solve the problem of motion planning under uncertainty: the Markov Decision Process (MDP) and the Partially Observable Markov Decision Process (POMDP). The former represents a decision-making scheme which assumes that the state is fully observable but the action outcome is affected by noise. The latter relaxes the fully-observability assumption and provides a scheme to reason in the belief state instead of in the state space.

7.4.1 Probabilistic Motion Planning

In a probabilistic framework, motion planning is the problem of choosing the right action to achieve the task, coping with the uncertainty present in the system. A good overview of the state of the art on the topic is available in

literature [63]. In this context, a reward function r is defined in order to quantify the associated gain (and cost) of a given action u executed at state x:

$$r = r(x, u). \tag{7.35}$$

We call policy π the function mapping all the past measurements and actions to the next action. Formally:

$$\pi : z_{1:t-1}, u_{1:t-1} \rightarrow u_t. \tag{7.36}$$

In case of fully-observable state, the policy is a function of the state only:

$$\pi : x_{1:t-1} \rightarrow u_t. \tag{7.37}$$

Here, we introduce the concept of cumulative reward, to account for the sum of the expected rewards achieved executing actions over a time horizon. In fact, the selected action is the one that maximises the cumulative reward function R_h considering the execution a finite number h of consecutive actions.

$$R_h = E\Big[\sum_{\tau} \nu^{\tau} r_{t+\tau}\Big]. \tag{7.38}$$

In literature, the parameter ν is referred to as discount factor. It is used to give more value to earlier than later rewards.

$$\nu \in [0; 1]. \tag{7.39}$$

7.4.2 Markov Decision Process

Markov Decision Process (MDP) provides a mathematical framework for solving decision-making problems in stochastic environments, optimising action selection over an infinite time horizon h. A few assumptions are made:

1. the state x is fully observable
2. the state x is finite
3. the set of available actions u for each state is finite

Under these assumptions, MDP provides optimal policies to map states to actions, as in Eq. 7.37. In the simplest, greedy case, i.e. when $T = 1$, the policy chooses the action that maximises the immediate payoff:

$$\pi(x) = \arg\max_{u} r(x, u). \tag{7.40}$$

For each policy, a so-called *value function* measures the expected cumulative reward for each of the states. In the greedy case:

$$V_1(x) = \nu \max_u(x, u). \tag{7.41}$$

For longer time horizons, the policy is calculated to maximise the expected cumulative reward over h. Hence:

$$\pi_h(x) = \arg\max_u \left\{ r(x,u) + \int V_{h-1}(x')p(x'|u,x)dx' \right\}. \tag{7.42}$$

And the corresponding value function takes the following form:

$$V_h(x) = \nu \max_u \left\{ r(x,u) + \int V_{h-1}(x')p(x'|u,x)dx' \right\}. \tag{7.43}$$

In the limit case of an infinite time horizon, the value function converges to reach an equilibrium called the Bellman Equation, expressed in Eq. (7.44).

$$V_\infty(x) = \nu \max_u \left\{ r(x,u) + \int V_\infty(x')p(x'|u,x)dx' \right\}. \tag{7.44}$$

In order to compute an optimal policy whose value function satisfies Eq. 7.44, a dynamic programming algorithm named *Value Iteration* [5] was firstly introduced in 1957. Since then, a number of alternative algorithms have been proposed, of which the most common is *Policy Iteration*. In algorithm 6, *Value Iteration* is presented in pseudo code.

Algorithm 6 MDP Value Iteration

1: **for** $i = 1 : N$ **do**
2: $V = V_0$ ▷ Initialisation of vector V, e.g. all zeros
3: **end for**
4: **while** V does not converge **do**
5: **for** $i = 1 : N$ **do**
6: $V(x_i) = \nu \max_u[r(x_i,u) + \sum_{j=1}^{N} V(x_j)p(x_j|u,x_i)]$ ▷ Iterative calculation as in Eq. 7.43
7: **end for**
8: **end while**
9: $\pi(x) = \arg\max_u[r(x,u) + \sum_{j=1}^{N} V(x_j)p(x_j|u,x_i)]$ ▷ Policy definition
10: **return** π

For the presented *Value Iteration* algorithm, a "synchronous" and an "asynchronous" update can be adopted for line 6. At t-ith iteration, the former calculates all $V(x_i)$ using the values of V at $t-1$, the latter performs the update of each element $V(x_i)$ one-by-one using the most-recent values of the vector. Both schemes guarantees the convergence:

$$V \to V_\infty. \tag{7.45}$$

7.4.3 Partially Observable Markov Decision Process

Partially Observable Markov Decision Process (POMDP)[9] provides a framework for action selection in the belief state. It assumes that:

1. the state x is finite
2. a belief bel is built on the state x
3. the set of available actions u for each state is finite
4. the time horizon h is finite.

POMDP generalises the decision-making scheme of MDP, dropping the assumption of full state observability. This implies that the next action is chosen based on the belief state, instead of the state itself. Hence, Eq. (7.43) and Eq. (7.42) are replaced by:

$$V_h(bel) = \nu \max_u \left\{ r(bel, u) + \int V_{h-1}(bel')p(bel'|u, x)d\, bel' \right\} \tag{7.46}$$

and

$$\pi_h = \arg\max_u \left\{ r(bel, u) + \int V_{h-1}(bel')p(bel'|u, x)d\, bel' \right\}. \tag{7.47}$$

Unlike MDP, in this case the value $V(bel)$ is defined as a function of an entire probability distribution. This makes the computation of Eq. (7.46) expensive, even for a low-dimension state vector x, and small time-horizon h. Pruning techniques were developed to reduce the number of linear constraints for such equation [8]. Though, the pure not-greedy POMDP algorithm remains computationally not tractable for most applications, and a number of approximated solutions were proposed to simplify the problem and overcome its intractability [63].

Bibliography

[1] Action inference - the next best touch. `http://vimeo.com/49121076`. Accessed: 2014-09-23. pages 18

[2] World relative trajectories pomdp. `http://people.csail.mit.edu/kjhsiao/wrtpomdps/index.html`. Accessed: 2014-09-23. pages 18

[3] Ahissar, E., and Knutsen, P. M. Object localization with whiskers. *Biological cybernetics 98*, 6 (2008), 449–458. pages 11

[4] Barragan, P. R., Kaelbling, L. P., and Lozano-Perez, T. Interactive bayesian identification of kinematic mechanisms. In *IEEE International Conference on Robotics and Automation* (May 2014), pp. 2013–2020. pages 94

[5] Bellman, R. A markovian decision process. *Indiana Univ. Math. J. 6* (1957), 679–684. pages 15, 129

[6] Boers, Y., Driessen, H., Bagchi, A., and Mandal, P. Particle filter based entropy. In *13th Conference on Information Fusion (FUSION)* (July 2010), pp. 1–8. pages 94, 95

[7] Bruyninckx, H., Soetens, P., and Koninckx, B. The real-time motion control core of the Orocos project. In *IEEE International Conference on Robotics and Automation* (2003), pp. 2766–2771. pages 117

[8] Cassandra, A., Littman, M. L., and Zhang, N. L. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* (1997), Morgan Kaufmann Publishers, pp. 54–61. pages 130

[9] Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. Acting optimally in partially observable stochastic domains, 1994. pages 15, 130

[10] Castano-cano, D., Grossard, M., and Hubert, A. Multi-axis force sensing with pre-stressed resonant composite plates: An alternative to strain gauge force sensors. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on* (2014), IEEE, pp. 1361–1367. pages 11

[11] Şahin, E., Çakmak, M., Doğar, M. R., Uğur, E., and Üçoluk, G. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 15*, 4 (Dec. 2007), 447–472. pages 91

[12] da Silva, J. G., de Carvalho, A. A., and da Silva, D. D. A strain gauge tactile sensor for finger-mounted applications. *Instrumentation and Measurement, IEEE Transactions on 51*, 1 (2002), 18–22. pages 10

[13] David, O., Russotto, F.-X., Simoes, M. D. S., and Measson, Y. Collision avoidance, virtual guides and advanced supervisory control teleoperation techniques for high-tech construction: framework design. *Automation in Construction 44*, 0 (2014), 63 – 72. pages 6, 11

[14] De Geeter, J., De Schutter, J., Bruyninckx, H., Van Brussel, H., and Decreton, M. Tolerance-weighted l-optimal experiment design for active sensing. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on* (Oct 1998), vol. 3, pp. 1670–1675 vol.3. pages xii, 12, 15, 39, 40

[15] De Geeter, J., Van Brussel, H., De Schutter, J., and Decreton, M. Recognising and locating objects with local sensors. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on* (Apr 1996), vol. 4, pp. 3478–3483 vol.4. pages 13, 19

[16] De Schutter, J., Bruyninckx, H., Dutré, S., De Geeter, J., Katupitiya, J., Demey, S., and Lefebvre, T. Estimating first-order geometric parameters and monitoring contact transitions during force-controlled compliant motion. *The International Journal of Robotics Research 18*, 12 (1999), 1161–1184. pages 59

[17] Doucet, A., De Freitas, N., and Gordon, N., Eds. *Sequential Monte Carlo methods in practice*. 2001. pages 12

[18] Erickson, L., Knuth, J., O'Kane, J. M., and LaValle, S. M. Probabilistic localization with a blind robot. In *Proc. IEEE International Conference on Robotics and Automation* (2008). pages 14

[19] Fox, C., Evans, M., Pearson, M., and Prescott, T. Tactile slam with a biomimetic whiskered robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (May 2012), pp. 4925–4930. pages 11

[20] Fox, D., Burgard, W., and Thrun, S. Active markov localization for mobile robots. *Robotics and Autonomous Systems 25* (1998), 195–207. pages 14

[21] Gadeyne, K., and Bruyninckx, H. Markov techniques for object localization with force-controlled robots. In *In Proc. of ICAR* (2001). pages xii, 11, 12, 13, 19, 40, 41, 65

[22] Gadeyne, K., Lefebvre, T., and Bruyninckx, H. Bayesian hybrid model-state estimation applied to simultaneous contact formation detection and geometrical parameter estimation. *Int. J. Robotics Research 24* (2005), 615–630. pages 13, 38

[23] Galdeano, D., Chemori, A., Krut, S., and Fraisse, P. Task-based whole-body control of humanoid robots with zmp regulation, real-time application to a squat-like motion. In *Multi-Conference on Systems, Signals Devices (SSD), 2014 11th International* (Feb 2014), pp. 1–6. pages 114

[24] Gaston, P. C., and Lozano-Perez, T. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6* (1983), 257–265. pages 11

[25] Gelb, A. *Applied optimal estimation*. MIT press, 1974. pages 12, 125

[26] Gentle, J. E., Härdle, W., and Mori, Y. *Handbook of Computational S tatistics*. Springer, 2004. pages 71

[27] Grimson, W., and Lozano-Pérez, T. Model-based recognition and localization from sparse range or tactile data, 1984. pages 11

[28] Hebert, P., Burdick, J., Howard, T., Hudson, N., and Ma, J. Action inference: The next best touch. *Robotics: Science and Systems Workshop on Mobile Manipulation* (2012). pages xiii, 10, 12, 13, 14, 16, 19, 38, 49, 50, 51, 58, 60, 65, 77, 83, 91, 92, 108

[29] Hsiao, K., Kaelbling, L. P., and Lozano-Pérez, T. Task-driven tactile exploration. Robotics: Science and Systems Conference. pages 16

[30] Hsiao, K., Kaelbling, L. P., and Lozano-Pérez, T. Robust grasping under object pose uncertainty. *Autonomous Robots 31*, 2-3 (2011), 253–268. pages 10, 12, 13, 38, 42, 51, 91

[31] Hsiao, K., Lozano-pérez, T., and Kaelbling, L. P. Robust belief-based execution of manipulation programs. In *Eighth Intl. Workshop on the Algorithmic Foundations of Robotics* (2008). pages 14, 17, 57, 91

[32] Hsiao, K., of Technology. Dept. of Electrical Engineering, M. I., and Science, C. *Relatively Robust Grasping*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2009. pages xii, 15, 19, 43, 51, 60, 62, 65, 83, 86, 94, 108

[33] Javdani, S., Klingensmith, M., Bagnell, D., Pollard, N., and Srinivasa, S. Efficient Touch Based Localization through Submodularity. *ArXiv e-prints* (Aug. 2012). pages 17

[34] Jung, D., and Zelinsky, A. Whisker based mobile robot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (1996), pp. 497–504. pages 11

[35] Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82*, Series D (1960), 35–45. pages 12, 124

[36] Kearns, M., Mansour, Y., and Ng, A. Y. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Learning in graphical models*. Springer, 1998, pp. 495–520. pages 61

[37] Kim, D., and Möller, R. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems 55*, 3 (2007), 229 – 243. pages 11

[38] Kitagaki, K., Suehiro, T., and Ogasawara, T. Monitoring of a pseudo contact point for fine manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (1996), vol. 2, IEEE, pp. 757–762. pages 11, 56, 119

[39] Kolesar, E. S. Piezoelectric tactile sensor, June 2 1998. US Patent 5,760,530. pages 10

[40] Kullback, S., and Leibler, R. On information and sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), 79–86. pages 14, 96

[41] Laugier, C., and Theveneau, P. Planning sensor-based motions for part-mating using geometric reasoning techniques. In *ECAI* (1986), pp. 617–629. pages 46

[42] Lefebvre, T., Bruyninckx, H., and De Schutter, J. Polyhedral contact formation modeling and identification for autonomous compliant

motion. *IEEE Transactions on Robotics and Automation 19*, 1 (Feb 2003), 26–41. pages 11, 13, 19

[43] Lefebvre, T., Bruyninckx, H., and De Schutter, J. Task planning with active sensing for autonomous compliant motion. *The International Journal of Robotics Research 24*, 1 (2005), 61. pages xii, 15, 44, 46, 48, 51, 96

[44] Lengagne, S., Vaillant, J., Yoshida, E., and Kheddar, A. Generation of whole-body optimal dynamic multi-contact motions. *The International Journal of Robotics Research 32*, 9-10 (2013), 1104–1119. pages 114

[45] Maiolino, P., Maggiali, M., Cannata, G., Metta, G., and Natale, L. A flexible and robust large scale capacitive tactile system for robots. *Sensors Journal, IEEE 13*, 10 (Oct 2013), 3910–3917. pages 11

[46] Maurelli, F., Krupinski, S., Petillot, Y., and Salvi, J. A particle filter approach for auv localization. In *OCEANS 2008* (Sept 2008), pp. 1–7. pages 13

[47] Merlhiot, X. A robust, efficient and time-stepping compatible collision detection method for non-smooth contact between rigid bodies of arbitrary shape. *Multibody Dynamics 9* (2007). pages 89

[48] Mihaylova, L., Lefebvre, T., Bruyninckx, H., Gadeyne, K., and Schutter, J. D. Active sensing for robotics - a survey. In *in Proc. 5 th Int'l Conf. On Numerical Methods and Applications* (2002), pp. 316–324. pages 93

[49] Miller, A. T., and Allen, P. K. Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE 11*, 4 (2004), 110–122. pages 112

[50] Mittendorfer, P., and Cheng, G. Humanoid multimodal tactile-sensing modules. *IEEE Transactions on Robotics 27*, 3 (2011), 401–410. pages 114

[51] Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics 24*, 1 (feb. 2008), 15 –26. pages 91

[52] Nikandrova, E., Laaksonen, J., and Kyrki, V. Towards informative sensor-based grasp planning. *Robotics and Autonomous Systems 62*, 3 (2014), 340 – 354. Advances in Autonomous Robotics, Bristol, {UK}. pages 10, 12, 17, 51, 95

[53] Nitti, D., De Laet, T., and De Raedt, L. Distributional clauses particle filter. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 504–507. pages 113

[54] O'Kane, J. M., and LaValle, S. M. Localization with limited sensing. *IEEE Transactions on Robotics 23* (August 2007), 704–716. pages 14

[55] Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics* (1962), 1065–1076. pages 95

[56] Petrovskaya, A., and Khatib, O. Global localization of objects via touch. *IEEE Transactions on Robotics 27*, 3 (June 2011), 569–585. pages xii, 14, 41, 42, 60, 65, 67, 77

[57] Petrovskaya, A., Khatib, O., Thrun, S., and Ng, A. Y. Bayesian estimation for autonomous object manipulation based on tactile sensors. In *IEEE International Conference on Robotics and Automation* (2006). pages 11, 12, 13, 16, 19, 38, 108

[58] Russell, R. A. Using tactile whiskers to measure surface contours. In *IEEE International Conference on Robotics and Automation* (1992), IEEE, pp. 1295–1299. pages 11

[59] Ryan, A., and Hedrick, J. K. Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems 58*, 5 (2010), 574 – 584. pages 13

[60] Shannon, C. E. A mathematical theory of communication. *Bell system technical journal 27* (1948). pages 16, 94

[61] Sotiropoulos, P., Tosi, N., Andritsos, F., and Geffard, F. Optimal docking pose and tactile hook-localisation strategy for auv intervention: The difis deployment case. *Ocean Engineering 46*, 0 (2012), 33 – 45. pages 11

[62] Taguchi, Y., Marks, T., and Hershey, J. Entropy-based motion selection for touch-based registration using rao-blackwellized particle filtering. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Sept 2011), pp. 4690–4697. pages 16, 95

[63] Thrun, S., Burgard, W., and Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. pages 12, 83, 94, 113, 125, 128, 130

[64] Thrun, S., Fox, D., Burgard, W., and Dellaert, F. Robust monte carlo localization for mobile robots. *Artificial intelligence 128*, 1 (2001), 99–141. pages 13

[65] Tosi, N., David, O., and Bruyninckx, H. Dof-decoupled active force sensing (d-dafs): A human-inspired approach to touch-based localisation tasks. In *IEEE International Conference on Advanced Robotics* (2013). pages 11

[66] Tosi, N., David, O., and Bruyninckx, H. Action selection for touch-based localisation trading off information gain and execution time. In *International Conference on Robotics and Automation* (2014). pages 51

[67] Wan, E., and Van der Merwe, R. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* (2000), pp. 153–158. pages 12, 125

[68] Zhu, F., and Spronck, J. A capacitive tactile sensor for shear and normal force measurements. *Sensors and Actuators A: Physical 31*, 1 (1992), 115–120. pages 10

Curriculum

Personal Data

Niccoló Tosi
Birth: 31st October 1985, Bagno a Ripoli, Florence, Italy
contact: niccolo.tosi@kuleuven.be or niccolo.tosi@gmail.com
webpage: http://people.mech.kuleuven.be/∼u0095674/

Education

- **2011 - current**: **PhD Student in Mechanical Engineering** at the Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium, in collaboration with CEA, France. Supervised by Prof. Herman Bruyninckx (KULeuven) and Mr. Olivier David (CEA).
- **2008 - 2009**: **Master of Science in Maritime Engineering Science**, University of Southampton, UK.
- **2004 - 2008**: **Bachelor of Engineering in Mechanical Engineering**, University of Florence, Italy.
- **1999 - 2004**: **High School**, Liceo Classico Galileo, Florence, Italy.

Work Experience

- **Nov 2010 - May 2011**: **Mobile App Developer**, self-employed.
- **Feb 2010 - Oct 2010**: **Marie Curie Early Stage Researcher**, Tactile Localisation for AUV trajectory correction in the framework of the Freesubnet Project.

List of Publications

Articles in Internationally Reviewed Journals

1. Niccoló Tosi, Olivier David and Herman Bruyninckx.
   DOF Decoupling Task Graph Model: Reducing the Complexity of Touch-based Active Sensing
   Manuscript under review.

2. Panagiotis Sotiropoulos, Niccoló Tosi, Fivos Andritsos and Franck Geffard.
   Optimal docking pose and tactile hook-localisation strategy for AUV intervention: The DIFIS deployment case
   Ocean Engineering, 46:33–45, June 2012.

Papers at International Conferences and Symposia, Published in Full in Proceedings

1. Niccoló Tosi, Olivier David and Herman Bruyninckx.
   Action Selection for Touch-based Localisation Trading Off Information Gain and Execution Time.
   In: *Proceedings of the 2014 IEEE International Conference on Robotics and Automation*, Hong Kong.

2. Niccoló Tosi, Olivier David and Herman Bruyninckx.
   DOF-Decoupled Active Force Sensing (D-DAFS): A human-inspired approach to touch-based localisation tasks.
   In: *Proceedings of the 2013 IEEE International Conference on Advanced Robotics*, Montevideo, Uruguay.

Workshop Presentations at International Conferences and Symposia

1. Niccoló Tosi, Olivier David and Herman Bruyninckx.
   DOF-Decoupled Active Sensing: A (more) human approach to localization tasks.
   In: *IEEE 2013 Conference on Robotics and Automation, Mobile Manipulation Workshop*, Karlsruhe, Germany.

Project Presentations and Reports

1. Enrico Di Lello, Herman Bruyninckx, Niccoló Tosi, Dominick Vanthienen
   Localization of the stop assembly button: an active sensing strategy
   Rosetta Project, report for project partners, 2012.

2. Niccoló Tosi, Olivier David and Herman Bruyninckx.
   Active Sensing For Object Localisation: State of The Art and Current Research
   Robohow Workshop, KTH, Stockholm, 2014.

Other Presentations

1. Niccoló Tosi, Olivier David and Herman Bruyninckx.
   Active Force Sensing For Object Localisation
   In: *Journée des Jeunes Chercheurs en Robotique*, Rennes, France, 2012.

Index

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF MECHANICAL ENGINEERING
PRODUCTION ENGINEERING, DESIGN AND AUTOMATION DIVISION
Celestijnenlaan 300B box 2420
B-3001 Heverlee
info@mech.kuleuven.be
www.mech.kuleuven.be